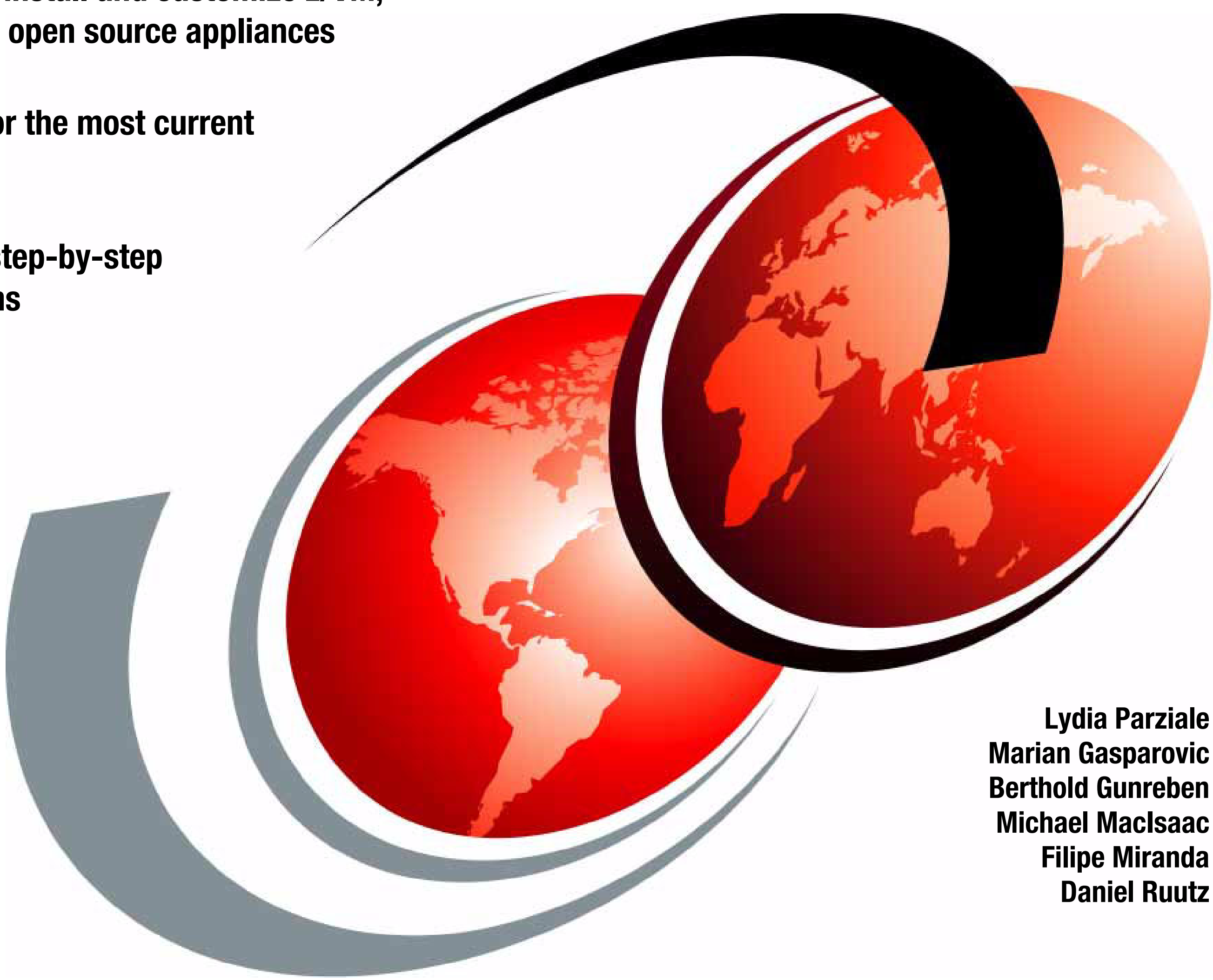
IBM

The Virtualization Cookbook for IBM z/VM 6.3, RHEL 6.4, and SLES 11 SP3

A guide to install and customize z/VM, Linux, and open source appliances

Updated for the most current releases

Detailed, step-by-step instructions



Lydia Parziale
Marian Gasparovic
Berthold Gunreben
Michael MacIsaac
Filipe Miranda
Daniel Ruutz

Redbooks

ibm.com/redbooks

International Technical Support Organization

The Virtualization Cookbook for IBM z/VM 6.3, RHEL 6.4, and SLES 11 SP3

January 2014

SG24-8147-00

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

First Edition (January 2014)

This edition applies to Version 6, Release 3 of IBM z/VM, Red Hat Enterprise Linux version 6.4, and SUSE Linux Enterprise Server 11 SP3.

Contents

Notices

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

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

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

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

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

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

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

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

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

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

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

Trademarks

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

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

- AIX®
- DirMaint™
- Domino®
- DS8000®
- ECKD™
- eServer™
- FICON®
- HiperSockets™
- IBM®
- Lotus®
- MVS™
- OMEGAMON®
- RACF®
- Redbooks®
- Redpaper™
- Redbooks (logo) ®
- S/390®
- System Storage®
- System z10®
- System z9®
- System z®
- Tivoli®
- Velocity™
- WebSphere®
- z/OS®
- z/VM®
- z10™
- z9®

The following terms are trademarks of other companies:

Intel, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

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

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

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

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

Preface

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

— Albert Einstein

This IBM® Redbooks® publication describes how to *roll your own* Linux virtual servers on IBM System z® hardware under IBM z/VM®. It adopts a cookbook format that provides a concise, repeatable set of procedures for installing and configuring z/VM 6.3 into a single system image (SSI), then installing and customizing Linux.

You need at least two IBM System z logical partitions (LPARs) with associated resources, z/VM 6.3 media, and either the Red Hat Enterprise Linux (RHEL) version 6.4 or the SUSE Linux Enterprise Server (SLES) version 11 SP3 distribution (or both).

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, and to get some virtual servers up and running in a short amount of time (days, not weeks or months).

Parts of this book

This book consists of the following parts:

- Part 1, "Introduction and z/VM" on page 1 introduces the entire system, describes z/VM, describes planning, setting up ancillary machines, then installation and configuration into a two member SSI with z/VM 6.3.
- Part 2, "Red Hat Enterprise Linux 6.4" on page 149 describes installing and customizing RHEL.
- Part 3, "SUSE Linux Enterprise Server 11 SP3" on page 257 describes installing and customizing SLES.
- Part 4, "Other topics" on page 365 includes chapters on the following subjects:
  - Live Guest Relocation (LGR) between SSI members
  - Configuring IBM DirMaint™, SMAPI and IBM RACF®
  - Monitoring z/VM and Linux
  - Miscellaneous "recipes"
- Appendix A, "References and cheat sheets" on page 507 includes references, cheat sheets, and lists all source code in the tar file that is associated with this book.
- Appendix B, "Additional material" on page 513.

Conventions

The following font conventions are used in this book:

`Monospace and bold`	Commands entered by the user on the command line
`monospace`	Linux file, directories, and commands
`MONOSPACE CAPITALS`	z/VM files, virtual machine and minidisk names, and commands

Monospace bold italics	Values used to test this book, such as TCP/IP addresses. They should be replaced with values correct for your enterprise.

The following command conventions are used in this book:

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

Operating system releases used

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

z/VM 6.3:	GA code, July 2013
RHEL 6.4:	GA code, December 2012
SLES 11 SP3:	GM code, June 2013

Authors

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

Lydia Parziale is a Project Leader for the ITSO team in Poughkeepsie, New York, with domestic and international experience in technology management, including software development, project leadership, and strategic planning. Her areas of expertise include Linux on System z and database management technologies. Lydia is a Certified IT Specialist with an MBA in Technology Management and has been employed by IBM over 25 years in various technology areas.

Marian Gasparovic is an IT Specialist working for the IBM Server and Technology Group (STG) in IBM Slovakia. After working as an IBM z/OS® administrator at a business partner, he joined IBM as a Storage Specialist. Later, he worked as a Field Technical Sales Specialist responsible for new workloads. He joined STG Lab Services and Training in 2010. His main areas of expertise are z/VM and Linux as well as zBX and Unified Resource Manager. He also writes Redbooks publications at the ITSO.

Berthold Gunreben is a Build Service Engineer at SUSE in Germany. He has 14 years of professional experience in Linux and is responsible for the administration of the mainframe system at SUSE. Besides his expertise with Linux on System z, he is also a Mainframe System Specialist certified by the European Mainframe Academy: `http://www.mainframe-academy.de`. His areas of expertise include High Availability on Linux; Realtime Linux; Automatic Deployments; Storage Administration on the IBM DS8000®; Virtualization Systems with Xen, KVM, and z/VM; and Documentation. He has written extensively on many of the SUSE manuals.

Michael MacIsaac is an IBM Lab-based Services professional. His areas of expertise include z/VM, Linux, and the solutions that run on them. He has written extensively on these topics.

Filipe Miranda is the Global Lead for Linux on IBM System z for Red Hat Inc. based in California. His key responsibility is to set the overall Linux on System z strategy inside Red Hat. He has more than 12 years of experience in Linux and Open Source technologies. He joined Red Hat Inc. seven years ago at the Red Hat Brazil branch where he worked as a

Senior Solution Architect for Latin America promoting ISV and HIV Alliances. He holds two degrees, one in Data Communication from University of California Los Angeles and another degree in Computer Science from the University Paulista in Sao Paulo, Brazil.

Daniel Ruutz is a Mainframe System Programmer and Manager at the Department of Defence Australia. He has 10 years of mainframe hardware and operating system experience. While designing and supporting the department's z/OS operating systems and hardware, he introduced a virtualized Linux capability to increase the return from the existing mainframe investment. He holds a mainframe-focused degree in Information Technology from Griffith University. He endeavors to create and promote more capable and affordable mainframe environments to his department and the broader IT community.

Special thanks

Special thanks goes out to the following people who performed usability tests:

Christopher Cramer, Susan Greenlee, Srivatsan Venkatesan, Eva Yan
IBM Poughkeepsie

Thanks to the following people for their contributions to this project:

Dave Bennin, Rich Conway, Roy P. Costa, Robert Haimowitz
International Technical Support Organization, Poughkeepsie Center

Alan Altmark, Bill Bitner, Sue Farrell, Tami Zebrowski-Darrow
IBM Endicott

Tracy Dean
IBM Software Group

Marcelo Moreira de Mello
Senior Software Maintenance Engineer, Red Hat Brazil

Jeremy Agee, Arlinton Bourne, Andrew Lemay, Darrell Rohwer
Red Hat

David Boyes
Sine Nomine Associates

Mark Post
SUSE

Thanks to many others in IBM Poughkeepsie and to the many who answered questions on the Linux-390 and IBMVM list servers.

Thanks to the authors of the previous editions of this book:

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

Now you can become a published author, too!

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

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

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

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

  ibm.com/redbooks

- Send your comments in an email to:

  redbooks@us.ibm.com

- Mail your comments to:

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

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks

- Follow us on Twitter:

  http://twitter.com/ibmredbooks

- Look for us on LinkedIn:

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

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

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

- Stay current on recent Redbooks publications with RSS Feeds:

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

Summary of changes

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

Summary of Changes
for SG24-8147-00
for The Virtualization Cookbook for IBM z/VM 6.3, RHEL 6.4, and SLES 11 SP3
as created or updated on December 13, 2014.

Summary of changes in this book

Many changes were made when this book was converted back into an IBM Redbooks publication, starting in June 2013:

- The z/VM version has been updated to 6.3.
- The Red Hat version has been updated to RHEL 6.4.
- The SUSE version has been updated to SLES 11 SP3.
- Section 2.8, “Planning worksheets” on page 20 and the following sections contain blank worksheets that have both been reworked to more closely show and list the values that will be needed as you build your environment.
- Chapter 7, “Install a z/VM non-SSI LPAR” on page 119 is new. Use this chapter if you are not going to use the z/VM single system image (SSI) feature.
- Section 11.3, “Create an LDAP server appliance” on page 214 has been updated to include TLS/SSL encryption, which is required for authentication with LDAP.
- Chapter 14, “Red Hat Network Satellite Server” on page 241 is new for Red Hat.
- Section 18.7, “Create an RPM” on page 336 is new for SUSE.
- Chapter 19, “Installing Linux with AutoYaST2” on page 339 is new for SUSE.
- Chapter 20, “Creating appliances with KIWI” on page 349 is new for SUSE.
- Any miscellaneous recipes related to disks were moved into the new Chapter 24, “Working with disks” on page 437. New sections include the following topics:
  - 24.5, “Add SCSI/FCP disks” on page 449 which includes “multipathing”
  - 24.6, “HyperPAV for Linux use” on page 466
- Any miscellaneous recipes related to networks were moved into the new Chapter 25, “Working with networks” on page 471. New sections include the following topics:
  - 25.2, “Adding CTCs to an SSI cluster” on page 473
  - 25.3, “Setting up a private interconnect” on page 476
  - 25.4, “Creating a HiperSockets device between Linux and z/OS” on page 477
  - 25.5, “Configuring a port group with LACP” on page 480
- Disabling cgroup memory in the following sections: 9.2.8, “Disable cgroup memory” on page 190 for Red Hat, and 16.4.7, “Disable cgroup memory” on page 300 for SLES. This simple step can result in up to 1% more free memory on each Linux virtual server.
- More detailed Monitoring chapter including MONWRITE data.

- LOGONBY for z/VM in section 22.5.6, “Configure LogonBy processing” on page 413 is new.
- The RACF SMF data unload utility in section 22.5.7, “Use the RACF SMF data unload utility” on page 415 is new.
- Section 26.7, “Setting up the Linux Terminal Server” on page 500 is new.
- Section 26.8, “Redefine command privilege classes” on page 503 is new.
- Section 26.9, “Use Crypto Express to seed /dev/random” on page 503 is new.

Part 1

Introduction and z/VM

This part of the book starts with an introduction, discusses planning, then describes z/VM installation into a two-node single system image (SSI) cluster, configuration, and servicing. Part 1 consists of the following chapters:

- Chapter 1, “Introduction to z/VM and Linux” on page 3 gives a brief introduction of the book.
- Chapter 2, “Planning” on page 11 describes how to plan hardware, software, and networking resources. It describes DASD labeling conventions used in the book and password planning. Worksheets are provided for the examples. Blank copies for your use are also provided.
- Chapter 3, “Configure a desktop machine” on page 35 describes how to set up a Windows desktop machine to access z/VM and Linux.
- Chapter 4, “Configure an NFS/FTP server” on page 45 describes how to set up a distributed server running Linux (or UNIX) to perform the initial Linux installations. Later, this machine can be retired.
- Chapter 5, “Install a z/VM SSI cluster” on page 57 shows how to install and configure z/VM 6.3 to create a two-member single system image (SSI) cluster.
- Chapter 6, “Service z/VM” on page 103 describes how to apply service to z/VM both in the form of programming temporary fixes (PTFs) and recommended service upgrades (RSUs).

There is also a chapter on how to install a non-SSI z/VM 6.3 system:

- Chapter 7, “Install a z/VM non-SSI LPAR” on page 119

1

Introduction to z/VM and Linux

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

— Albert Einstein

Virtualization is *still* hot in the IT industry. The IBM mainframe, z/VM, and its predecessors have been doing virtualization for five different decades. Today, it can be argued that the mainframe is the most functionally rich virtualization platform. When Linux came to the IBM mainframe in 2000, it was a natural fit to run under z/VM. You can run many tens or even low hundreds of Linux virtual servers on the same System z logical partition (LPAR) under z/VM.

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

The following capabilities are some of the best strengths of the mainframe and z/VM:

- Their virtualization capabilities are more mature and robust than any other hardware and hypervisor combination.
- The z/VM virtual switch makes networking Linux much simpler.
- Full volume backup of systems allows for complete disaster recovery when another data center is available.
- z/VM is one of the easiest operating systems to customize at the base installation level. There is only a relatively small number of configuration files. Properly set up, z/VM can run for months with little maintenance or administration required.

Much function has been added to z/VM since version 5.2. The following brief summary provides the function added in the last three releases.

z/VM 6.3

z/VM 6.3, which became generally available in late July 2013, extends the mainframe virtualization platform to help you reshape and derive more value from your systems. It has been designed to offer the following benefits:

- Improved economies of scale with z/VM support for 1 TB of real memory
  - Better performance for larger virtual machines
    - Quadruples memory scalability while continuing to maintain greater than 90% resource utilization
  - Additional vertical scalability to help reduce LPAR sprawl
    - Considerably more virtual machines can be consolidated into a single LPAR, depending on workload characteristics
  - Reduced administrative expense through managing a smaller number of large-capacity z/VM host servers
- Improved performance with z/VM HiperDispatch
- More efficient utilization of CPU hardware resources for dispatched work
- IBM adopted OpenStack as part of its cloud strategy. In concert, IBM is making contributions to the OpenStack project that are designed to enable z/VM 6.3 to be the first System z operating environment to be managed by these open cloud architecture-based interfaces
- Simplified migration to z/VM V6.3 with upgrade in place, which reduces the effect of an upgrade on active workloads
- Highly secure industry-standard support that is required for banking and financial-industry applications
- Support for the new IBM zEC12 and zBC12 servers

z/VM 6.2

z/VM 6.2, which became generally available in December of 2011, continues to help clients extend business value across the enterprise by integrating applications and data while providing high levels of availability, security, and operational ease. This release implements multisystem virtualization of up to four z/VM systems. This new technology extends z/VM virtualization to a new level, which enables members of the cluster to share resources and synchronize. This gives the appearance of being a single system image (SSI).

With the IBM z/VM single system image feature (VMSSI), a running Linux virtual machine can be relocated non-disruptively from one member system to any other member, a process known as *live guest relocation* (LGR). This provides application continuity across planned z/VM and hardware outages.

Members of a cluster are part of the same Inter-System Facility for Communications (ISFC) collection, and use ISFC channel connections to communicate. Multiple channel-to-channel devices provide a greater capability for data to flow between members. All members of a cluster share direct access storage device (DASD) for virtual machines and selected z/VM data. Sharing minidisks between members improves the integrity and performance of the system and provides a foundation for live guest relocation.

Members of a z/VM SSI cluster are managed, serviced, and administered as one system. Resources including the user directory, minidisks, spool files, and network devices, used by both control program (CP) and virtual machines, will be shared among all members. Sharing of resources helps allow Linux guests access to the same devices and networks regardless of which member they are logged on to or where they are relocated.

Each member of a z/VM SSI cluster is able to communicate with other active members. When a z/VM system is configured as a member of a cluster, it automatically *joins* the other members during system startup. Coordination of members joining and leaving the cluster, maintaining a common view of member and resource states, and negotiating access to shared cluster resources are all accomplished in a seamless fashion. This coordination allows Linux guests to be relocated between members during planned outages. Linux guests can now be moved from one member to another during most planned outages (service upgrades) without interruption. This allows the Linux application to have continuous run time during planned outages, and therefore allows the application to experience no downtime.

To use the functions that define and maintain an SSI cluster, the VMSSI must be licensed and enabled. Servicing in an SSI cluster is simplified by using a single service stream for all members. Sharing service resources allows service to be rolled out to each member of the cluster on individual schedules, avoiding an outage for the entire cluster. This allows uninterrupted Linux guest availability because the Linux guest may be relocated to a different member before a planned outage.

z/VM 6.1

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

Enhancements in z/VM V6.1 provide the following benefits:

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

Read more about System z virtualization capabilities at the following website:

http://www.vm.ibm.com

z/VM 5.4

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

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

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

z/VM 5.3

z/VM 5.3 became generally available in June 2007. Scalability was extended to allow 256 GB of real memory, a total of 8 TB of virtual storage, and 32 real processors. z/VM V5.3 also added support for *Collaborative Memory Management Assist* (CMMA) on the IBM z9® EC and the z9 BC processors or later. *Virtual Machine Resource Manager* (VMRM) detects when memory is constrained and notifies the Linux guests, which can then adjust their memory consumption to help relieve the memory constraint. In the previous major release, z/VM 5.2, many memory contention issues were removed with the control program (CP) now using memory above 2 GB for a much broader set of operations. Previously, guest pages had to be moved below 2 GB for many reasons, for example, in both standard I/O and queued direct I/O (QDIO). Now I/O can be 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.

1.1 What is virtualization?

Virtualization is the ability for a computer system to share resources so that one physical server can act as many *virtual servers*. z/VM allows the sharing of the mainframe's physical resources such as disk (DASD), memory (sometimes called *storage*), network adapters (OSA cards), and CPU (CPs or IFLs). These resources are managed by z/VM's hypervisor, which is also known as the *control program* (*CP*). When the user logs on to z/VM, the hypervisor creates a virtual machine, which can run one of many different operating systems. The two operating systems that are discussed in this book are the *Conversational Monitor System* (*CMS)* (which can be thought of as a z/VM *shell*), and *Linux*. Virtual machines running Linux as guests of z/VM become *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*. This book will use the latter, with the aim to use the same clear and insightful presentation.

Many books and papers are talking about virtualization today, but not telling you how to do it. The remainder of this book gives you the *HOW TO* that back up these marketing words.

1.3 Choices and decisions made in this book

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

- Use of a commercial Systems Management product: Because this is a book designed for you to learn the basics, the use of a commercial systems management product is not described.
- Directory Maintenance product versus the `USER DIRECT` file: Usually, editing the `USER DIRECT` file and using the **`DIRECTXA`** command is recommended over a directory maintenance product, such as the IBM *DirMaint* or CA *VM:Direct* products. However, Chapter 22, “DirMaint, SMAPI, and RACF” on page 371 was added to show how a directory maintenance product can be used.
- Shared read-only Linux `/usr/` file system versus read/write: Some cloning solutions use an environment that shares the `/usr/` file system. This choice often makes the solution more complex, especially when adding software to the virtual servers. A read/write `/usr/` file system on the virtual servers is chosen to keep things as simple as possible.
- Conventional 3390 IBM ECKD™ DASD versus FBA disks accessed with SCSI over FCP: 3390 ECKD DASD is described, however, section 24.5, “Add SCSI/FCP disks” on page 449 has been added to address more disk types.
- Cloning script or EXEC versus manual installation: Two methods of cloning are described: Manually and with a Linux bash script. The *manual* method is described so that you can better learn the concepts. The *Linux* script is provided so you can save time.

1.4 Single system image design

With the introduction of z/VM 6.2 in December 2011, the architecture of Linux solutions on System z has changed dramatically. It is true that Cross Systems Extensions (CSE) allowed for a type of clustering environment for Linux on System z before z/VM 6.2. However, CSE was not widely used nor was the architecture completely enabled for clusters. z/VM 6.2, with its single system image (SSI) and live guest relocation (LGR) functions has changed all of this. No longer is it true that a z/VM system is the most important “object” in the hierarchy. With z/VM 6.1 and earlier, the system identifier of each z/VM system was the most important. With z/VM 6.2 and later, the SSI name is the highest level identifier.

A block diagram of a four member SSI, with default volume labels, is shown in Figure 1-1 on page 8. As is the recommend scenario, a four member cluster, with two members on two different central electronic complexes (CECs), is shown. In such a cluster, there are four z/VM systems and four system identifiers. However, there is only one SSI name. In this book, a two member SSI installed onto one CEC is described.

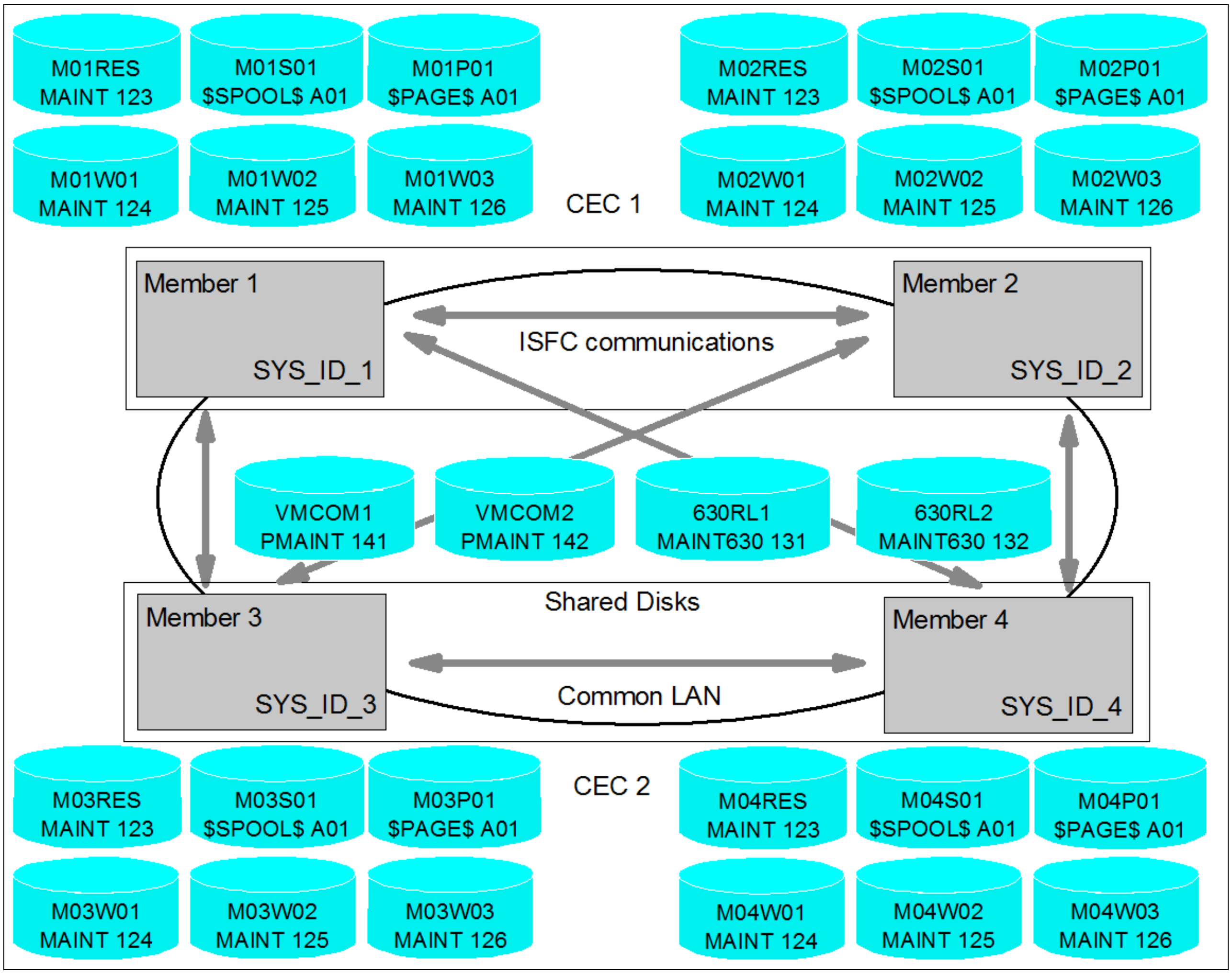


Figure 1-1 z/VM 6.3 single system image block diagram

1.5 Infrastructure design

To install and configure z/VM, and to install, configure, and clone Linux or *provision virtual servers*, there must be a certain infrastructure design in place. A System z server with associated resources and the z/VM operating system define much of this infrastructure. Figure 1-2 on page 9 shows a block diagram of a z114 with z/VM 6.2 on two LPARs. z/VM comes with many virtual machines predefined. The most important six IDs are shown in the z/VM LPAR above the dashed line. Below the dashed line, you see the virtual machines described in this book.

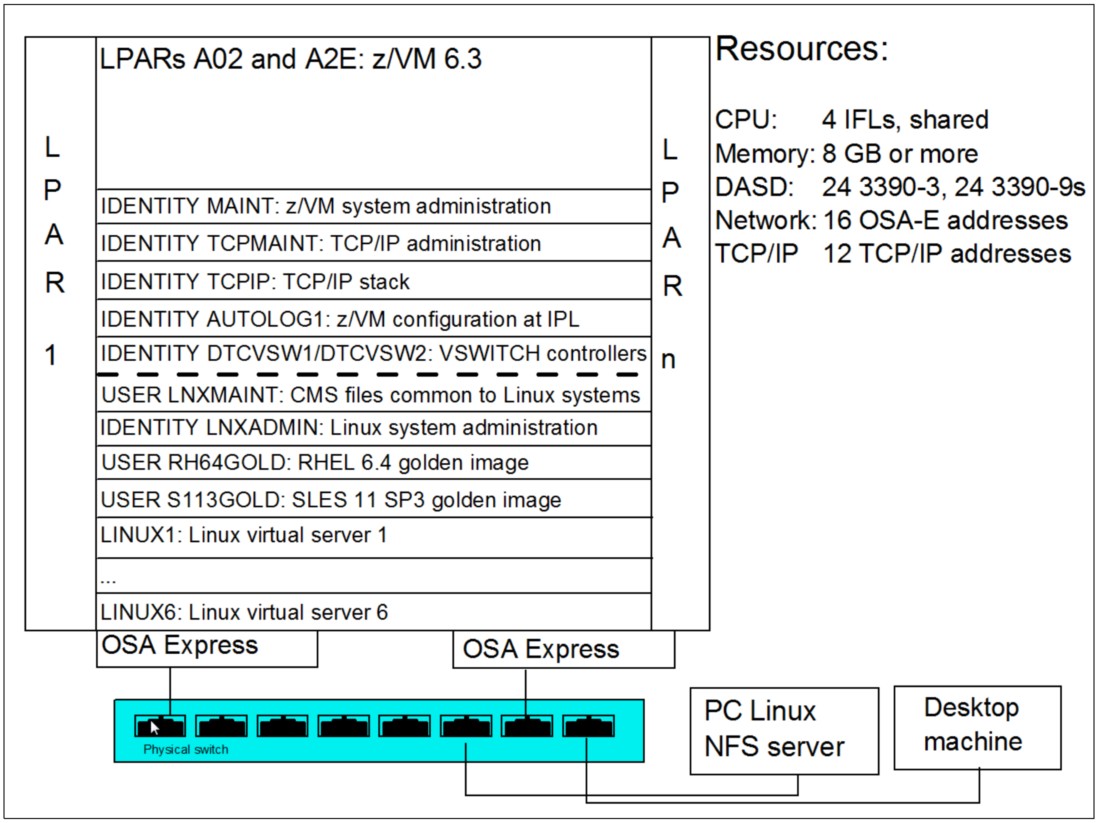


Figure 1-2 System infrastructure and z/VM virtual machines

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

LNXMAINT A virtual machine on which to store CMS files to be used by Linux

LNXADMIN The Linux system administration server that exports install trees, clones systems, and performs other system administrative functions. This is an *identity* that can be logged on to all SSI members at the same time

RH64GOLD A virtual machine that contains the RHEL 6.4 *golden image*

S113GOLD A virtual machine that contains the SLES 11 SP3 *golden image*

LINUX1 - LINUX6 Six sample *worker* virtual machines

In addition to the two LPARs, two other machines are shown:

Linux PC NFS server A Linux box used for the first two installations of each distribution

Desktop machine A workstation from where all of the work is done

1.6 Usability tests performed

During the years of writing of this book, many usability tests have been conducted. The participants had a variety of skills, but none had both solid Linux and z/VM system administration skills. By the end of two days, most participants had cloned their first Linux virtual server. You should be able to complete the steps in the book in four solid days of work, if all goes well and you work hard.

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 and Linux. It begins by describing planning for a z/VM single system image (SSI), then a *bill of materials*, or all the resources that you need. Next, it describes conventions that are adopted for labeling 3390 volumes. Finally, resource worksheets are presented for the following resources:

- z/VM resources
- Linux resources
- Linux virtual machines

2.1 Planning for an SSI and LGR

With z/VM 6.2 and 6.3, Linux systems can be relocated between logical partitions (LPARs) and CECs in the SSI cluster. This feature is called *live guest relocation* (LGR).

There are several reasons why you might need to relocate a running virtual server, for example, for workload rebalancing, software, or hardware maintenance. Before you relocate a guest, there are architectural, disk, memory, and networking requirements that you must understand. Below are some hints to help with installation of the VMSSI feature, and tips to get you started relocating a Linux guest.

2.1.1 Hints and tips

Keep in mind that even if you have previous experience with installation and service of z/VM, it is important that you read the instructions for installation of z/VM 6.3 with or without the VMSSI feature. To plan and prepare for z/VM 6.3 in advance of the general availability, you are encouraged to use the publications: *z/VM: Getting Started with Linux on System z*, SC24-6194-03, and chapter 25 of *z/VM: CP Planning and Administration*, SC24-6178-05.

An SSI cluster must have direct logical links between all systems. All SSI clusters use Inter-System Facility for Communications (ISFC) for intra-cluster communication for LGR. ISFC uses channel-to-channel (CTC) devices. For maximum throughput, when you are setting up your network, follow the section: “Guidelines for planning your network in an SSI cluster”, in chapter 2 of *z/VM: Getting Started with Linux on System z*. Faster CTC speeds increase throughput and result in shorter relocations.

Factors that can affect relocation

The following factors should be considered in planning for Linux LGR:

- Virtual machine memory: The size and use of the virtual machine's memory can affect relocation performance. Parts of the processing for relocation are proportional to the size of the virtual machine. The cost of this processing increases with larger virtual machines. Relocation performance is also impacted by the frequency and amount of memory being changed in the virtual machine.
- Matching virtual machine configurations: To prepare for live guest relocation, ensure that the virtual machine has a configuration that allows for it to be relocated and that a matching configuration can be set up on the destination member. For information about configuration requirements and about verifying a virtual machine's eligibility to relocate, refer to the manual, *z/VM: CP Planning and Administration*, chapter 27.
- CPU utilization: The z/VM V6.3 single system image will synchronize all the members in the cluster. You must ensure that you have allocated enough system resources to account for the necessary synchronization and communication among members. After initialization, the synchronization overhead is relatively low. Communication between members does increase during negotiations for access to devices and other resources, as well as during live guest relocation. For example, two independent systems that run fine at peak utilization (close to 100%), when joined in a cluster may have performance problems.

  For z/VM members that are running as a second-level z/VM system, they should not be waiting for CPU more than 10% of the time. For more information, refer to the “Resource Limit Conditions” section of the manual, *z/VM: CP Planning and Administration*, chapter 27.
- Paging and other system resources: To prepare for live guest relocation, the target system must have enough system resource during and after the relocation. You will need to ensure that your paging space is adequate. To be safe, there should be twice as much

space available as the total virtual memory that can be defined on the system. The easiest way to check on this aspect of system resources is to issue the **CP QUERY ALLOC PAGE** command, which will show the percent used, the slots available, and the slots in use. If you add in the size of the virtual machines being relocated (a 4 KB page = a 4 KB slot) to the slots in use, and that brings the in-use percentage over 50%, that can have an undesirable impact on system performance. Remember that this query command provides only a snapshot in time.

- Real memory: Real memory resources are important for both the source and the destination systems for relocations. You will need enough real memory (1) to hold buffers during the relocation on both systems, and (2) to accommodate the incoming guest's working set afterward on the target system. Relocation performance will also be affected by the level of overall resource constraint for both the source and destination systems.
- Linux distributions and live guest relocations: With the introduction of LGR among members of your SSI, it is increasingly important to identify the level of Linux on System z that is running within each member. The latest level of a distribution release is considered supported by the Linux Distribution Partners. The best practice for setting up VMSSI is to ensure that you are running on the latest level and that your distribution is supported by your Linux distributor.

2.1.2 Need for ECKD DASD

If z/VM 6.3 is to be installed into an SSI, at least one extended count key data (ECKD) volume is necessary for the Persistent Data Record (PDR).

If RACF is planned to be implemented in an SSI, the database must be configured as being shared and at least two ECKD direct access storage device (DASD) volumes are necessary. Concurrent virtual and real reserve/release must always be used for the RACF database DASD when RACF is installed in an SSI. See the *z/VM: RACF Security Server System Programmer's Guide* for more information about sharing a RACF database, and *z/VM: CP Planning and Administration* for information on DASD Sharing.

2.2 Bill of materials

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

- Hardware
- Software
- Networking

2.2.1 Hardware resources

The following hardware is needed:

- One, two, or four System z LPARs:
  - Processors or CPUs per LPAR: One IFL (or CP) minimum; two or more are recommended
  - Memory: 8 GB main storage or more. With z/VM 6.3, expanded storage is no longer recommended. For more information, see the following website:

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

   - DASD: 24 3390-3s and 24 3390-9s were allocated for the reference system described in this book
   - Open Systems Adapter (OSA) network cards: One card minimum with six device numbers. Two OSA Express cards with six device numbers are recommended for high availability
- A network-attached computer that will act as an NFS server and possibly an FTP server with at least 6 GB of disk space
- A workstation or desktop machine that has network access to the mainframe

2.2.2 Software resources

The following software resources are needed:

- z/VM 6.3 installation media with documentation. The physical media of DVDs is described. If you will be using *Shopz* to download the z/VM installation media and make it available utilizing an FTP server, physical media will not be needed.

  > **Note:** FTP servers running on Microsoft Windows are not supported; therefore, Linux or UNIX are recommended.

- RHEL 6.4 Linux installation media. If you do not have it, you can request a 180-day evaluation copy at no charge from the following website:

  `http://www.redhat.com/z`

  See section 4.3, “Set up an NFS server” on page 49 for details.
- SLES 11 SP3 Linux ISO images. For an evaluation copy, see the following SUSE website:

  `http://www.suse.com/products/server/eval.html`

  Then, drill down on the **System z** link.
- An operating system for the NFS server: The same Linux distribution you will use on System z is recommended.
- The code associated with this book, which is available at the following website:

  `http://www.vm.ibm.com/devpages/mikemac/SG248147.tgz`
- Tools on the workstation and desktop:
   - A 3270 Emulator such as *Attachmate Extra*, Hummingbird *Host Explorer*, or IBM *Personal Communications* for Windows desktops
   - A Linux Secure Shell (SSH) client such as PuTTY
   - A Virtual Network Computing (VNC) viewer such as RealVNC

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

2.2.3 Networking resources

The following network resources are needed:

- TCP/IP addresses for each z/VM SSI member
- One TCP/IP address for each Linux virtual server
- Associated TCP/IP information:
   - DNS host name

- DNS domain name
- DNS server TCP/IP address
- TCP/IP gateway
- TCP/IP subnet mask
- TCP/IP MTU size

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

2.3 z/VM conventions

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

2.3.1 Volume labeling convention

You should have a convention for labeling DASD. Your shop might already have a labeling convention that will largely determine the labels to be given to the DASD used by your z/VM LPAR(s).

Each System z DASD has a real device address consisting of four hexadecimal digits. Each System z DASD has a six character label. It is convenient to include the four-digit address in the label so that you can easily tell the address of each DASD from its label. When followed thoroughly, this convention guarantees that no two DASDs will have the same label. This can be an important issue, especially when a z/OS LPAR 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 or SSI that *owns* the DASD. Therefore, the volume labeling convention used in this book identifies the LPAR or SSI with the first character. That leaves the second character in the label to identify the basic function of the DASD. The example SSI in this book is identified by the character “J”.

The first character in the label

The letter “J” is hardcoded into the **CPFORMAT** REXX EXEC in the file associated with this book, which can be found in Appendix B, “Additional material” on page 513. This EXEC utilizes this volume labeling convention. If you want a different LPAR identifier character, they can easily be changed (search for the `firstChar` variable). Following is the pertinent line of code:

```
/***********************************************************/
...
  Address COMMAND
  firstchar = 'J'
...
```

The second character in the label

The following characters are used for the types of DASD in the second character of the label:

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

2.3.2 Backup file naming convention

It is recommend that you keep copies of important z/VM and Linux configuration files. You should always keep copies of original configuration files in case you need to go back to them. Because z/VM file names are limited to 16 characters (eight for the file name and eight for the file type), only the last four characters of the file type are used. This often requires some characters to be overwritten. For the original file, the suffix `ORIG` is used, and for the most recent working copy, the suffix `WRKS` (for “it WoRKS”!) is used. For example, the original `USER DIRECT` file is copied to the file `USER DIREORIG` before it is modified the first time, then later copied to the file `USER DIREWRKS`. In this fashion, there is always a copy of the original, the current, and the last working copy of configuration files.

2.3.3 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 that you are looking for. The default Linux shell, *bash*, does this by default with the up arrow and down arrow keys.

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

2.4 Disk planning

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

- Conventional ECKD DASD versus fixed-block architecture (FBA) disks over Small Computer System Interface-Fibre Channel Protocol (SCSI-FCP)
- Size of 3390 disks: Mod-3s, Mod-9s, Mod-27s, or larger
- Amount of disk storage per Linux image and how to allocate file systems

DASD versus SCSI-FCP

This book describes how to use conventional ECKD DASD and only briefly discusses FBA disks accessed over SCSI-FCP. This is not because either technology is superior, but because DASD seems to be much common than SCSI/FCP disks, and it is easier to virtualize given that SCSI/FCP disks require worldwide port name/logical unit number (WWPN/LUN) identifiers. If you were to use SCSI/FCP disks, cloning with the **clone.sh** script would have to be modified to account for these identifiers, and other reasons. Sometimes, a combination of these two types of disk storage is used. When this 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, 3390-9s, or larger

Emulated 3390-3s format to about 2.3 GB, while 3390-9s (6.8 GB) and 3390-27s (20 GB) are each three times larger. z/VM 6.3 can be installed on to either 3390-3s or 3390-9s. Some large System z shops are moving to volumes larger than 3390-27s to avoid hitting the 64 K limit of real device addresses (four-character hexadecimal).

Disk storage per Linux image

Disk storage has the following characteristics:

- This version of the book now recommends one 3390-9 DASD split in half to create minidisks at virtual addresses `100` and `101`. This gives about 6.8 GB of space. You may choose to give the `100` and `101` minidisks each a full 3390-9, which would double the disk space to 13.6 GB.
- The root file system is on `/dev/dasda1`. It is not a logical volume; therefore, if there are any problems with Logical Volume Manager (LVM), the system is still able to boot.
- Other file systems are on logical volumes that are part of single volume group.

> **Important:** Regardless of how you choose to lay out the minidisks, it is important that the golden image and all target Linux virtual machines have two minidisks of the same size at virtual addresses `100` and `101`. This assumption is coded into the **`clone.sh`** script.

Swapping

z/VM has a feature called *virtual disks*, or *VDISKs*, that exist in memory but are presented to guest operating systems as disks. Memory is many, many times faster than disk, so using these for swap spaces makes sense.

Ideally, your Linux systems should never have to swap, but workloads cannot be predicted this easily. Therefore, all Linux virtual servers should have an adequate set of swap spaces. What defines “an adequate set of swap space” can turn into a debate. However, there seems to be a consensus in the z/VM and Linux community that one or two small swap spaces on virtual disk, which may or may not be also backed up by a larger swap space on real disk, is best. This book describes how to set up two VDISK swap spaces, but not the additional physical disk.

To create the swap spaces, the **`SWAPGEN EXEC`** is used. It creates and formats a Linux swap disk from CMS. The current version is available for download on the web at the following site:

http://download.sinenomine.net/swapgen

Instructions for downloading and installing `SWAPGEN` can be found in the `swapgen-readme.txt` file on the preceding swapgen web page.

2.5 Memory planning

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

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

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

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

- The IBM z/VM Performance Resource pages:

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

- The IBM z/VM web page on configuring processor storage:

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

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

2.6 Password planning

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

This book considers different system administration roles:

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

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

The method of backing up z/VM data onto the Linux administration system means that the Linux administrator will have access to all z/VM passwords. Therefore, the examples in this book set all z/VM and Linux system administration passwords to the same value, `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 might want to define a finer granularity for passwords based on the following system administration roles:

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

The sets of passwords that you define depends on the roles that your organization adopts.

2.7 CTC planning

It is important to plan adequate channel-to-channel (CTC) definitions to achieve an adequate live guest relocation (LGR) quiesce and relocation time. At a minimum, it is recommended that two CTC devices are connected per SSI member via two channel paths. During the SSI installation process, you can install only two CTCs per SSI member. After installation, see section 25.2, “Adding CTCs to an SSI cluster” on page 473, to add CTC devices.

To decide which CTCs to install initially, you will need input/output definition file (IODF) information from your hardware configuration colleague. They will need to provide two FICON Native CTC (FCTC) control units that each have a minimum of four devices.

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

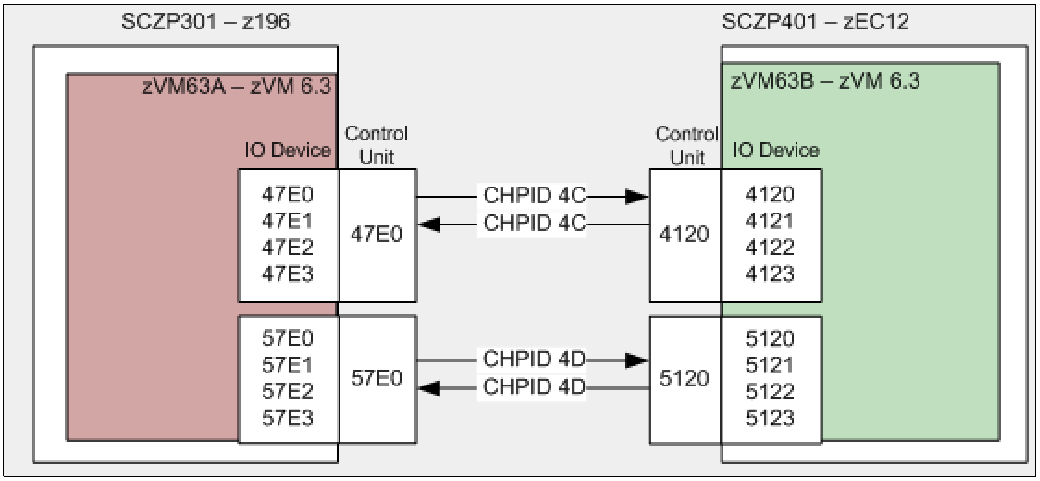


Figure 2-1 FCTCs between the IBM z196 (SCZP301) and zEC12 (SCZP401)

Example 2-1 shows sample IODF configuration statements that represent the FCTC connections.

Example 2-1 Sample IODF configuration statements

SSI member 1

```
CNTLUNIT CUNUMBR=47E0,PATH=((CSS(0),4C)),UNITADD=((00,004)),   *
       LINK=((CSS(0),0E)),CUADD=2E,UNIT=FCTC
IODEVICE ADDRESS=(47E0,004),UNITADD=00,CUNUMBR=(47E0),         *
       STADET=Y,PARTITION=((CSS(0),A02)),UNIT=FCTC
CNTLUNIT CUNUMBR=57E0,PATH=((CSS(0),4D)),UNITADD=((00,004)),   *
       LINK=((CSS(0),0A)),CUADD=2E,UNIT=FCTC
IODEVICE ADDRESS=(57E0,004),UNITADD=00,CUNUMBR=(57E0),         *
       STADET=Y,PARTITION=((CSS(0),A02)),UNIT=FCTC
```

SSI member 2

```
CNTLUNIT CUNUMBR=4120,PATH=((CSS(2),4C)),UNITADD=((00,004)),   *
       LINK=((CSS(2),31)),CUADD=2,UNIT=FCTC
IODEVICE ADDRESS=(4120,004),UNITADD=00,CUNUMBR=(4120),         *
       STADET=Y,PARTITION=((CSS(2),A2E)),UNIT=FCTC
CNTLUNIT CUNUMBR=5120,PATH=((CSS(2),4D)),UNITADD=((00,004)),   *
       LINK=((CSS(2),30)),CUADD=2,UNIT=FCTC
IODEVICE ADDRESS=(5120,004),UNITADD=00,CUNUMBR=(5120),         *
    STADET=Y,PARTITION=((CSS(2),A2E)),UNIT=FCTC
```

From the provided CTC information, the selected CTC devices from ***zVM63A*** will be ***47E0*** and ***57E0***. For ***zVM63B*** devices, ***4120*** and ***5120*** will be used.

More information on CTC capacity recommendations can be found at the following site:

http://www2.marist.edu/~mvmua/13011100.pdf

2.7.1 z/VM resources used in this book

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

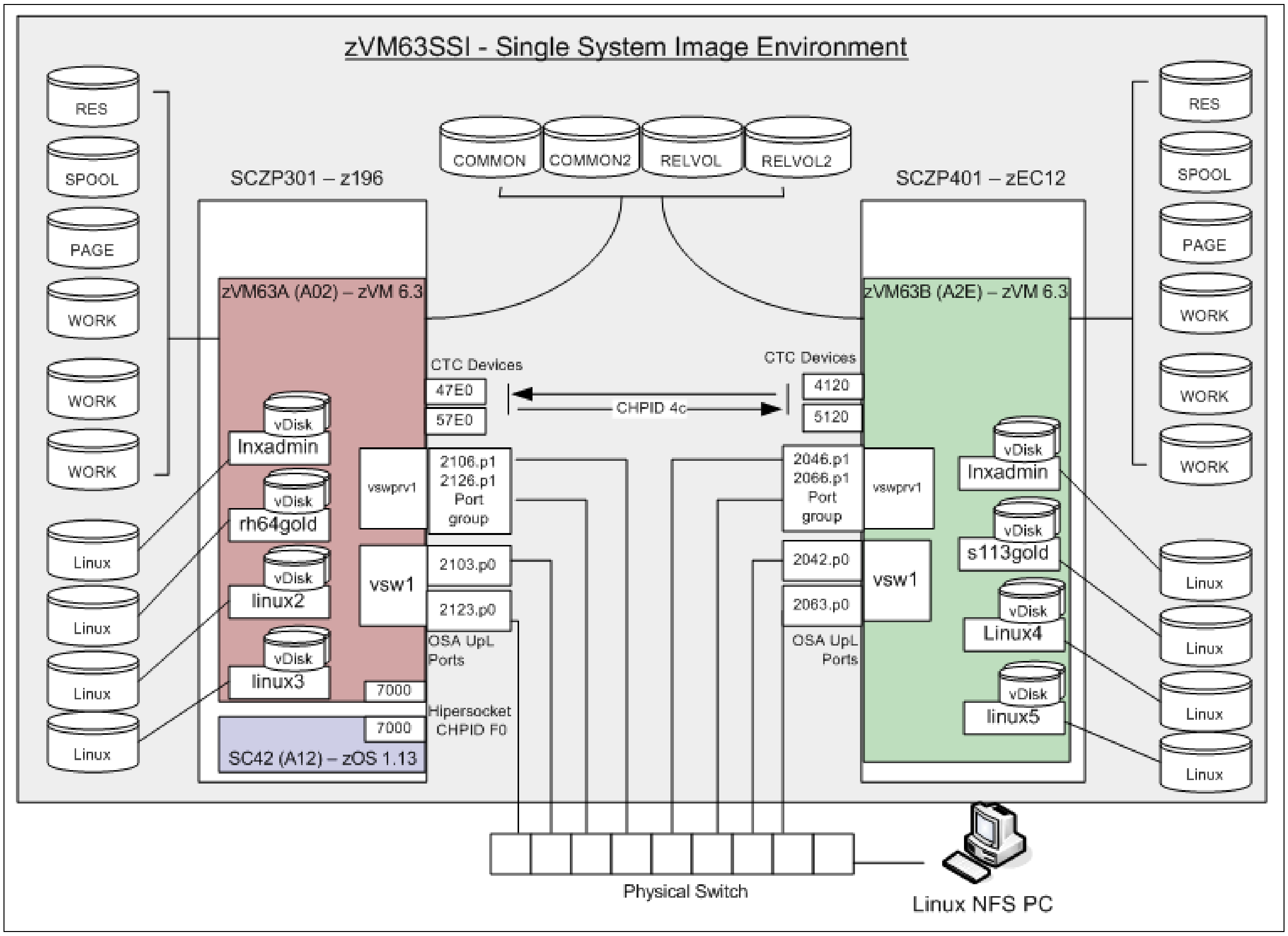


Figure 2-2 Mainframe environment used in this book

2.8 Planning worksheets

Worksheets are included in this section. They are populated with the resources used in writing this book. There are also corresponding blank worksheets in section 2.9, "Blank planning worksheets" on page 28.

> **Important:** Some values in tables that follow are shown in ***monospace bold italics***. This font convention is used to signify that you should replace the example value with the correct value for your site. This convention is used throughout the book.

2.8.1 IBM Shopz worksheet

If you are ordering z/VM using Shopz as described in section 5.1, "Obtain z/VM through electronic download" on page 58, use the blank worksheet found in Table 2-13 on page 29 to document the values that you will use. Table 2-1 provides information about the values that are required for each row.

Table 2-1 Shopz worksheet

Name	Value	Comment
Starting URL	http://www.ibm.com/software/ShopzSeries	
User ID	***MyCustomerNumber***	
Password	***MyPassword***	
Order number		Write down for reference

2.8.2 HMC worksheet

Section 5.3.1, "Start the z/VM installation" on page 61 describes how to start a z/VM installation from the Hardware Management Console (HMC). Complete the worksheet in Table 2-14 on page 29 to document the values that you will use. Table 2-2 provides information about the values that are required for each row.

Table 2-2 HMC values worksheet

Name	Value	Comment
HMC location or URL	ITSO URL	Physical HMC access or through the web?
HMC user ID	***MyUserID***	
HMC password	***MyPassword***	
FTP source system	***9.12.5.251***	If installing z/VM from an FTP server
z/VM installation directory	`/srv/ftp/zvm6.3`	

2.8.3 Worksheets for the INSTPLAN command

INSTPLAN worksheet 1

Section 5.3.2, "Copy a vanilla z/VM system to DASD" on page 63 describes the **INSTPLAN** command run from the Integrated 3270 Console. Complete the worksheet in Table 2-15 on page 29 to document the values that you will use on the first two panels. Table 2-3 provides information about the values required for each row.

Table 2-3 INSTPLAN values worksheet for first two panels

Name	Value	Comment
Language	AMENG	AMENG (American English), USENG (uppercase English), or KANJI
DASD model	3390-3	3390 Model-3 or Model-9 (FBA disk is not described in this book)
File pool name	pool1	
System type	SSI and non-SSI	Both installation types described in different chapters
Non-SSI system name	***ZVM63C***	Non-SSI system identifier

Name	Value	Comment
Number of members	2	For SSI installation only (usually 2 or 4)
SSI cluster name	*ZVM63SSI*	For SSI installation only
Automatic configuration	no	Will z/VM be maintained automatically? “No” is strongly recommended

INSTPLAN worksheet 2

Complete the worksheet in Table 2-16 on page 30 to document the values that you will use on the third installation panel described in section 5.3.2, “Copy a vanilla z/VM system to DASD” on page 63. The member names will become the z/VM system identifiers, and the LPAR names should be the same names as on the HMC. Table 2-4 provides information about the values required for each row.

Table 2-4 INSTPLAN values worksheet for panel 3

Slot	Member name	LPAR name	Comment
1	*ZVM63A*	*A02*	Member 1 system identifier and LPAR name
2	*ZVM63B*	*A2E*	Member 2 system identifier and LPAR name
3			Member 3 system ID and LPAR name (optional)
4			Member 4 system ID and LPAR name (optional)

INSTPLAN worksheet 3

Complete the worksheet in Table 2-17 on page 30 to document the volume labels and real device addresses that you will use on the third installation panel that is described at the end of section 5.3.2, “Copy a vanilla z/VM system to DASD” on page 63. Table 2-5 provides information about the values that are required for each row.

Table 2-5 INSTPLAN values worksheet for volume definition

Type	Label	Address	Comment
COMMON	*JV1036*	*1036*	Common volume 1
COMMON2	*JV1037*	*1037*	Common volume 2
RELVOL	*JV1136*	*1136*	Release volume 1
RELVOL2	*JV1137*	*1137*	Release volume 2
Mem 1 RES	*JV1030*	*1030*	Member 1 residence volume
Mem 1 SPOOL	*JS1031*	*1031*	Member 1 spool volume
Mem 1 PAGE	*JP1032*	*1032*	Member 1 page volume
Mem 1 WORK	*JV1033*	*1033*	Member 1 work volume 1
Mem 1 WORK	*JV1034*	*1034*	Member 1 work volume 2 (3390-3 only)
Mem 1 WORK	*JV1035*	*1035*	Member 1 work volume 3 (3390-3 only)
Mem 2 RES	*JV1130*	*1130*	Member 2 residence volume
Mem 2 SPOOL	*JS1131*	*1131*	Member 2 spool volume

Type	Label	Address	Comment
Mem 2 PAGE	*JP1132*	*1132*	Member 2 page volume
Mem 2 WORK	*JV1133*	*1133*	Member 2 work volume 1
Mem 2 WORK	*JV1134*	*1134*	Member 2 work volume 2 (3390-3 only)
Mem 2 WORK	*JV1135*	*1135*	Member 2 work volume 3 (3390-3 only)
Mem 3 RES			Member 3 residence volume (optional)
Mem 3 SPOOL			Member 3 spool volume (optional)
Mem 3 PAGE			Member 3 page volume (optional)
Mem 3 WORK			Member 3 work volume 1 (optional)
Mem 3 WORK			Member 3 work volume 2 (optional)
Mem 3 WORK			Member 3 work volume 3 (optional)
Mem 4 RES			Member 4 residence volume (optional)
Mem 4 SPOOL			Member 4 spool volume (optional)
Mem 4 PAGE			Member 4 page volume (optional)
Mem 4 WORK			Member 4 work volume 1 (optional)
Mem 4 WORK			Member 4 work volume 2 (optional)
Mem 4 WORK			Member 4 work volume 3 (optional)

INSTPLAN worksheet 4

The values in Table 2-6 document the common volume and CTC addresses used in this book. This pane is shown at the end of section 5.3.2, “Copy a vanilla z/VM system to DASD” on page 63.

If you have only two members in the SSI, you need to specify only two pairs of CTCs (from member 1 to member 2, and vice versa).

Table 2-6 INSTPLAN values worksheet for volume definition

Real addresses for the common volume on each member LPAR:			
Member 1	**Member 2**	**Member 3**	**Member 4**
CTC device addresses:			
From member 1		**From member 2**	
To: member 1	N/A	To: member 1	*4120* *5120*
To: member 2	*47E0* *57A0*	To: member 2	N/A
To: member 3	________ ________	To: member 3	________ ________
To: member 4	________ ________	To: member 4	________ ________
From member 3		**From member 4**	
To: member 1	________ ________	To: member 1	________ ________

To: member 2	_______ _______	To: member 2	_______ _______
To: member 3	N/A	To: member 3	_______ _______
To: member 4	_______ _______	To: member 4	N/A

2.8.4 z/VM networking resources used in this book

Table 2-7 lists the networking resources used in the examples in this book. They are needed when invoking the **IPWIZARD** and when creating a VSWITCH for the Linux virtual machines.

Table 2-7 z/VM and networking resources worksheet

Name	Value	Comment
TCP/IP user ID	TCPIP	TCPIP is recommended
z/VM host name, member 1	***virtcook11***	
z/VM host name, member 2	***virtcook12***	
TCP/IP domain name	***itso.ibm.com***	System domain name usually set in DNS
TCP/IP gateway	***9.12.4.1***	The router to and from the local subnet
DNS server 1	***9.12.6.7***	Assigned by the network administrator
DNS server 2/3		Optional
Interface name	eth0	
OSA starting device number	***2040***	Start of OSA *triplet* for z/VM TCP/IP stack
Subnet mask	***255.255.240.0***	Assigned by network administrator
OSA device type	QDIO (layer 3)	
MTU size	1500	Check with network administrator
Primary OSA device for virtual switch	***4203***	Specify the first real device number and the next two device numbers will also be used
Secondary OSA device for virtual switch	***4300***	Ideally, it should be on a different CHPID/OSA card

2.8.5 z/VM DASD used in this book

Table 2-8 lists the z/VM DASD resource values that are used in the examples in this book.

Table 2-8 z/VM DASD used in this book

Device number	Label	Type	Notes
1030	***JV1030***	CP owned (3390-3)	Member 1 system residence volume
1031	***JS1031***	CP owned (3390-3)	Member 1 spool volume 1
1032	***JP1032***	CP owned (3390-3)	Member 1 paging volume 1
1033	***JV1033***	CP owned (3390-3)	Member 1 work volume 1
1034	***JV1034***	CP owned (3390-3)	Member 1 work volume 2

Device number	Label	Type	Notes
1035	*JV1035*	CP owned (3390-3)	Member 1 work volume 3
1036	*JV1036*	CP owned (3390-3)	Common volume 1
1037	*JV1037*	CP owned (3390-3)	Common volume 2
1038	*JV1038*	CP owned (3390-3)	Member 1 Paging volume 2
1039	*JV1039*	CP owned (3390-3)	Extend MAINT630 disks - not needed with GA
103A	*JV103A*	CP owned (3390-3)	Member 1 Paging volume 3
103B	*JR103B*	System (3390-3)	RACF database volume 1
1130	*JV1130*	CP owned (3390-3)	Member 2 system residence volume
1131	*JS1131*	CP owned (3390-3)	Member 2 spool volume 1
1132	*JP1132*	CP owned (3390-3)	Member 2 paging volume 1
1133	*JV1133*	CP owned (3390-3)	Member 2 work volume 1
1134	*JV1134*	CP owned (3390-3)	Member 2 work volume 2
1135	*JV1135*	CP owned (3390-3)	Member 2 work volume 3
1136	*JV1136*	CP owned (3390-3)	Common release volume 1
1137	*JV1137*	CP owned (3390-3)	Common release volume 2
1138	*JV1138*	CP owned (3390-3)	Member 2 Paging volume 2
1139	*JV1139*	CP owned (3390-3)	Member 2 Paging volume 3
113A	*JM113A*	System (3390-3)	LNXADMIN 200 on member 2 - for kiwi
113B	*JR113B*	System (3390-3)	RACF database volume 2
1260	*JM1260*	System (3390-9)	For adding a logical volume
1261	*JM1261*	System (3390-9)	For extending a logical volume
1262	*JM1262*	System (3390-9)	LNXMAINT 191-192, LNXADMIN 101 on member 1
1263	*JM1263*	System (3390-9)	LNXADMIN 100 on member 1
1264	*JM1264*	System (3390-9)	RH64GOLD 100 and 101
1265	*JM1265*	System (3390-9)	S113GOLD 100 and 101
1266	*JM1266*	System (3390-9)	LINUX1 100 and 101
1267	*JM1267*	System (3390-9)	LINUX2 100 and 101
1268	*JM1268*	System (3390-9)	LINUX3 100 and 101
1269	*JM1269*	System (3390-9)	LINUX4 100 and 101
126A	*JM126A*	System (3390-9)	LINUX5 100 and 101
126B	*JM126B*	System (3390-9)	LINUX6 100 and 101
1360	*JM1360*	System (3390-9)	LNXADMIN 200 on member 1(Satellite)
1361	*JM1361*	System (3390-9)	LNXADMIN 201 on member 1 (Satellite)

Device number	Label	Type	Notes
1362	*JM1362*	System (3390-9)	LNXADMIN 100 on member 2
1363	*JM1363*	System (3390-9)	LNXADMIN 101 on member 2
1364	*JM1364*	System (3390-9)	LNXADMIN 201 on member 2 - for kiwi
1365	*JM1365*	System (3390-9)	LNXADMIN 202 on member 1(Satellite)
1366	*JM1366*	System (3390-9)	LNXADMIN 203 on member 1(Satellite)
1367	*JM1367*	System (3390-9)	MONWRITE 191 and possibly others
1368	*JM1368*	System (3390-9)	For HYPERPAV (can be recycled if needed)
1369	*JM1369*	System (3390-9)	LINUX8 100 and 101
136A	*JM136A*	System (3390-9)	LNXADMIN 205 on member 1(Satellite)

Table 2-9 lists the z/VM DASD resource values that are used in a non-SSI installation.

Table 2-9 z/VM DASD used in this book: non-SSI

Device number	Label	Type	Notes
126C	*KV126C*	CP owned (3390-9)	System residence volume
126D	*KS126D*	CP owned (3390-9)	Spool volume
126E	*KP126E*	CP owned (3390-9)	Paging volume
126F	*KV126F*	CP owned (3390-9)	Work volume
136B	*KV136B*	CP owned (3390-9)	LNXADMIN 100
136C	*KV136C*	CP owned (3390-9)	Common volume
136D	*KV136D*	CP owned (3390-9)	Rel volume
136E	*KV136E*	CP owned (3390-9)	Additional paging volume
136F	*KV136F*	CP owned (3390-9)	LNXMAINT 191-192, LNXADMIN 101

2.8.6 Linux resources used in this book

Table 2-10 lists the Linux distributed NFS server resources that are used for the first System z Linux installation.

Table 2-10 Linux NFS server resources used in this book

Name	Value	Comment
TCP/IP address	***9.12.5.251***	
User/password	root/lnx4vm	
NFS-exported installation directory	/var/nfs/rhel64/ and /var/nfs/sles11sp3/	Directory with DVD 1 of each distribution

Table 2-11 lists the Linux resources that are used in the examples in this book.

Table 2-11 Linux resources used in this book

Name	Value	Comment
Linux root password	lnx4vm	
TCP/IP gateway	***9.12.4.1***	Obtain from network administrator
Subnet mask	***255.255.240.0***	Obtain from network administrator
DNS server	***9.12.6.7***	Obtain from network administrator
VNC installation password	12345678	Must be eight characters

2.8.7 Host names and IP addresses used in this book

Table 2-12 lists the host names and associated IP addresses that are used in the examples in this book.

Table 2-12 Hosts that are used in this book

Host name	IP address	Virtual machine	Notes
virtcook1.itso.ibm.com	***9.12.7.1***	LINUX1	Linux clone used with RHEL 6.4
virtcook2.itso.ibm.com	***9.12.7.2***	LINUX2	Linux clone used with RHEL 6.4
virtcook3.itso.ibm.com	***9.12.7.3***	LINUX3	Linux clone used with RHEL 6.4
virtcook4.itso.ibm.com	***9.12.7.4***	LINUX4	Linux clone used with SLES 11 SP 3
virtcook5.itso.ibm.com	***9.12.7.5***	LINUX5	Linux clone used with SLES 11 SP 3
virtcook6.itso.ibm.com	***9.12.7.6***	LINUX6	Linux clone used with SLES 11 SP 3
virtcook7.itso.ibm.com	***9.12.7.7***	LNXADMIN	Linux admin server for RHEL 6.4
virtcook8.itso.ibm.com	***9.12.7.8***	LNXADMIN	Linux admin server on SLES 11 SP3
virtcook9.itso.ibm.com	***9.12.7.9***	RH64GOLD	RHEL 6.4 “golden image”
virtcook10.itso.ibm.com	***9.12.7.10***	S113GOLD	SLES 11 SP3 “golden image”
virtcook11.itso.ibm.com	***9.12.7.11***	LPAR A02	z/VM 6.3 SSI member 1
virtcook12.itso.ibm.com	***9.12.7.12***	LPAR A2E	z/VM 6.3 SSI member 2
virtcook13.itso.ibm.com	***9.12.4.201***	LPAR A12	z/VM 6.3 non-SSI member 1

2.9 Blank planning worksheets

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

You might want to start by using the block diagram that is shown in Figure 2-3 to label the CECs, LPARs, and volume labels that you will use for your SSI cluster.

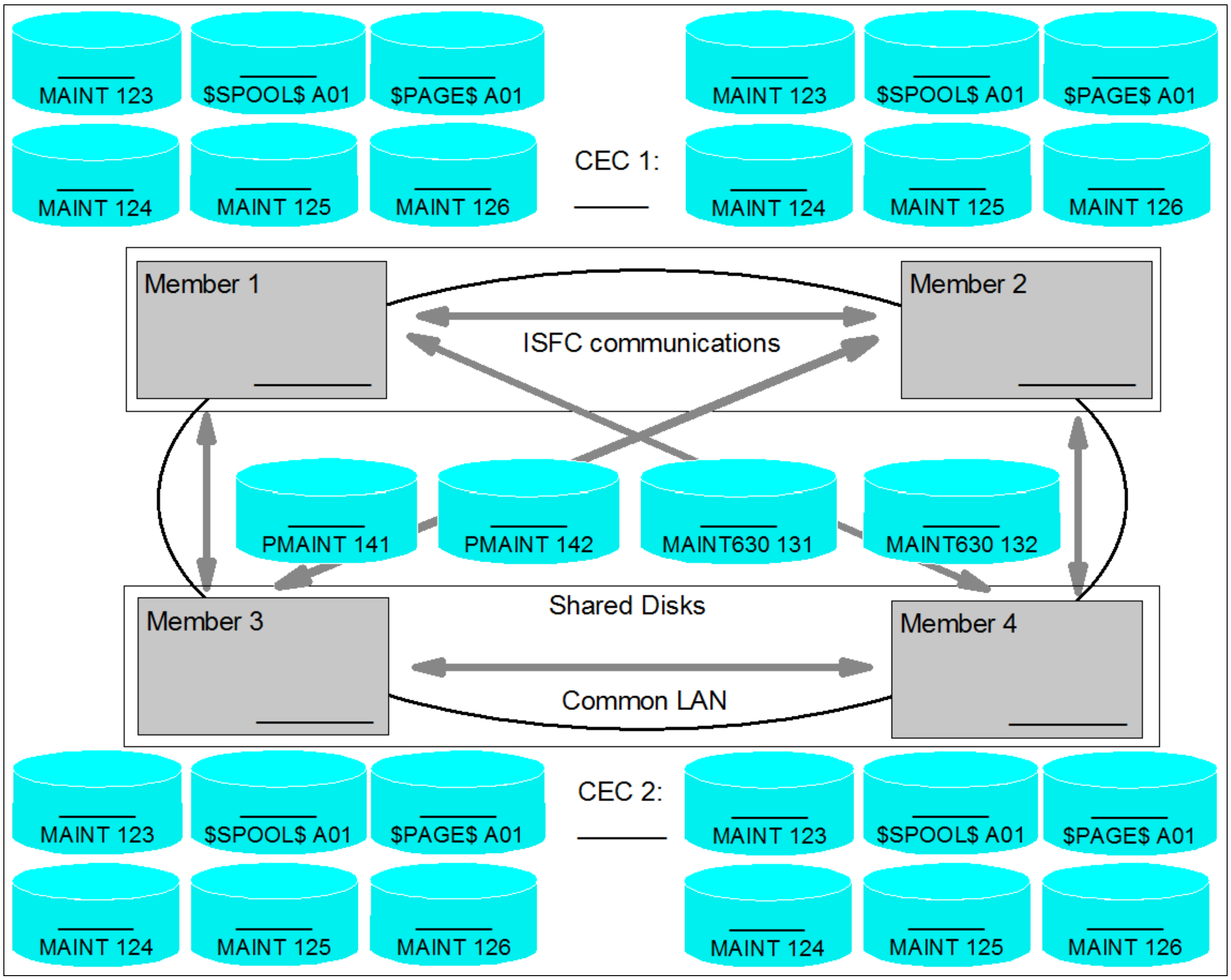


Figure 2-3 SSI block diagram blank worksheet

The following blank worksheets are included for your convenience. They are hopefully organized to be in the order that you will need them. It is recommended that you specify all values that will apply to make your installation process go more smoothly.

2.9.1 Shopz worksheet

If you are ordering z/VM using *Shopz*, as described in section 5.1, “Obtain z/VM through electronic download” on page 58, use the worksheet in Table 2-13 on page 29 to document the values that you will use.

Table 2-13 Shopz worksheet

Name	Value	Comment
Starting URL		
User ID		Customer number (for IBM employees, it is your intranet user ID and password)
Password		
Order number		

2.9.2 HMC worksheet

Section 5.3.1, "Start the z/VM installation" on page 61 describes how to start a z/VM installation from the HMC. Complete the worksheet in Table 2-14 to document the values that you will use.

Table 2-14 HMC values worksheet

Name	Value	Comment
HMC location or URL		Via the web
HMC user ID		
HMC password		
FTP source system		If installing z/VM from an FTP server
z/VM installation directory		`/srv/ftp/zVM6.3/`

2.9.3 INSTPLAN worksheets

Section 5.3.2, "Copy a vanilla z/VM system to DASD" on page 63, describes the INSTPLAN command run from the Integrated 3270 Console. The following information will be necessary.

INSTPLAN worksheet 1

Complete the worksheet in Table 2-15 to document the values used in the first two panels.

Table 2-15 INSTPLAN values worksheet for first two panels

Name	Value	Comment
Language		AMENG (American English), USENG (uppercase English), or KANJI
DASD model		3390 Model-3 or Model-9 (FBA disk is not described in this book)
File pool name		
System type		SSI or Non-SSI
Non-SSI system name		System identifier for non-SSI installation only
Number of members		For SSI installation only (usually 2 or 4)

Name	Value	Comment
SSI cluster name		For SSI installation only
Automatic configuration		Will z/VM be maintained automatically. “No” is strongly recommended

INSTPLAN worksheet 2

Complete the worksheet in Table 2-16 to document the values that you will use on the third installation panel that is described in section 5.3.2, “Copy a vanilla z/VM system to DASD” on page 63. The member names will become the z/VM system identifiers, and the LPAR names should be the same names as on the HMC.

Table 2-16 INSTPLAN values worksheet for panel 3

Slot	Member name	LPAR name	Comment
1			Member 1 system identifier and LPAR name
2			Member 2 system identifier and LPAR name
3			Member 3 system ID and LPAR name (optional)
4			Member 4 system ID and LPAR name (optional)

INSTPLAN worksheet 3

Complete the worksheet in Table 2-17 to document the volume labels and real device addresses that you will use on the third installation panel that is described near the end of section 5.3.2, “Copy a vanilla z/VM system to DASD” on page 63.

Table 2-17 INSTPLAN values worksheet for volume definition

Type	Label	Address	Comment
COMMON			Common volume 1
COMMON2			Common volume 2
RELVOL			Release volume 1
RELVOL2			Release volume 2
Mem 1 RES			Member 1 residence volume
Mem 1 SPOOL			Member 1 spool volume
Mem 1 PAGE			Member 1 page volume
Mem 1 WORK			Member 1 work volume 1
Mem 1 WORK			Member 1 work volume 2 (3390-3 only)
Mem 1 WORK			Member 1 work volume 3 (3390-3 only)
Mem 2 RES			Member 2 residence volume
Mem 2 SPOOL			Member 2 spool volume
Mem 2 PAGE			Member 2 page volume
Mem 2 WORK			Member 2 work volume 1
Mem 2 WORK			Member 2 work volume 2 (3390-3 only)

Type	Label	Address	Comment
Mem 2 WORK			Member 2 work volume 3 (3390-3 only)
Mem 3 RES			Member 3 residence volume (optional)
Mem 3 SPOOL			Member 3 spool volume
Mem 3 PAGE			Member 3 page volume
Mem 3 WORK			Member 3 work volume 1
Mem 3 WORK			Member 3 work volume 2 (3390-3 only)
Mem 3 WORK			Member 3 work volume 3 (3390-3 only)
Mem 4 RES			Member 4 residence volume (optional)
Mem 4 SPOOL			Member 4 spool volume
Mem 4 PAGE			Member 4 page volume
Mem 4 WORK			Member 4 work volume 1
Mem 4 WORK			Member 4 work volume 2 (3390-3 only)
Mem 4 WORK			Member 4 work volume 3 (3390-3 only)

INSTPLAN worksheet 4

Complete the worksheet in Table 2-18 to document common volume and CTC addresses. This panel is shown at the end of section 5.3.2, “Copy a vanilla z/VM system to DASD” on page 63.

If you have only two members in the SSI, you need to specify only two pairs of CTCs (from member 1 to member 2, and vice versa)

Table 2-18 INSTPLAN values worksheet for volume definition

Real addresses for the common volume on each member LPAR:			
Member 1	**Member 2**	**Member 3**	**Member 4**
CTC device addresses:			
From member 1		**From member 2**	
To: member 1	N/A	To: member 1	_______ _______
To: member 2	_______ _______	To: member 2	N/A
To: member 3	_______ _______	To: member 3	_______ _______
To: member 4	_______ _______	To: member 4	_______ _______
From member 3		**From member 4**	
To: member 1	_______ _______	To: member 1	_______ _______
To: member 2	_______ _______	To: member 2	_______ _______
To: member 3	N/A	To: member 3	_______ _______
To: member 4	_______ _______	To: member 4	N/A

2.9.4 z/VM Networking resources

Complete the worksheet in Table 2-19 to list the networking resources that will be needed when invoking the **IPWIZARD** and when creating a VSWITCH for the Linux virtual machines.

Table 2-19 z/VM and networking resources worksheet

Name	Value	Comment
TCP/IP user ID		`TCPIP` is recommended
z/VM host name, member 1		
z/VM host name, member 2		
TCP/IP domain name		System domain name usually set in DNS
TCP/IP gateway		The router to and from the local subnet
DNS server 1		Assigned by the network administrator
DNS server 2/3		Optional
Interface name		
OSA starting device number		Start of OSA *triplet* for z/VM TCP/IP stack
Subnet mask		Assigned by network administrator
OSA device type		
MTU size		Check with network administrator
Primary OSA device for virtual switch		Specify the first real device number and the next two device numbers will also be used
Secondary OSA device for virtual switch		Ideally, it should be on a different CHPID/OSA card

2.9.5 z/VM DASD worksheet

Use the worksheet in Table 2-20 to document the z/VM DASD that you will use.

Table 2-20 z/VM DASD blank worksheet

Device number	Label	Type	Notes

Device number	Label	Type	Notes

2.9.6 Linux resources worksheet

Use the worksheet in Table 2-21 to document the resources associated with the NFS server that will be used to be the installation source of the first System z Linux.

Table 2-21 Linux NFS server resources blank worksheet

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

Use the worksheet in Table 2-22 to document your System z Linux resources.

Table 2-22 Linux resources blank worksheet

Name	Value	Comment
Linux installation password		
Linux root password		
Linux TCP/IP gateway		
Linux TCP/IP broadcast		
Linux DNS server		
VNC installation password		

2.9.7 Host names and IP addresses worksheet

Use the worksheet in Table 2-23 to document the host names and associated IP addresses and virtual machines that you will use.

Table 2-23 Host names blank worksheet

Host name	IP address	Virtual machine/ LPAR	Notes

Host name	IP address	Virtual machine/ LPAR	Notes

3

Configure 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:

- A Secure Shell (SSH) client: PuTTY is recommended
- A Virtual Network Computing (VNC) client: RealVNC is recommended
- A 3270 emulator: Many choices are available

3.1 PuTTY: A free SSH client for MS Windows

Throughout this book, SSH is used to log in to Linux systems. It is easy to use and cryptographically secure. If you are using a Windows desktop, you will need a good SSH client. *PuTTY* is perhaps the most commonly used. You can download PuTTY from the following download web page:

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

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

Open PuTTY, and the configuration window that is shown in Figure 3-1 should open. If you spend a few minutes to configure PuTTY it may pay off in time savings. The examples shown below are using PuTTY Release 0.60:

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

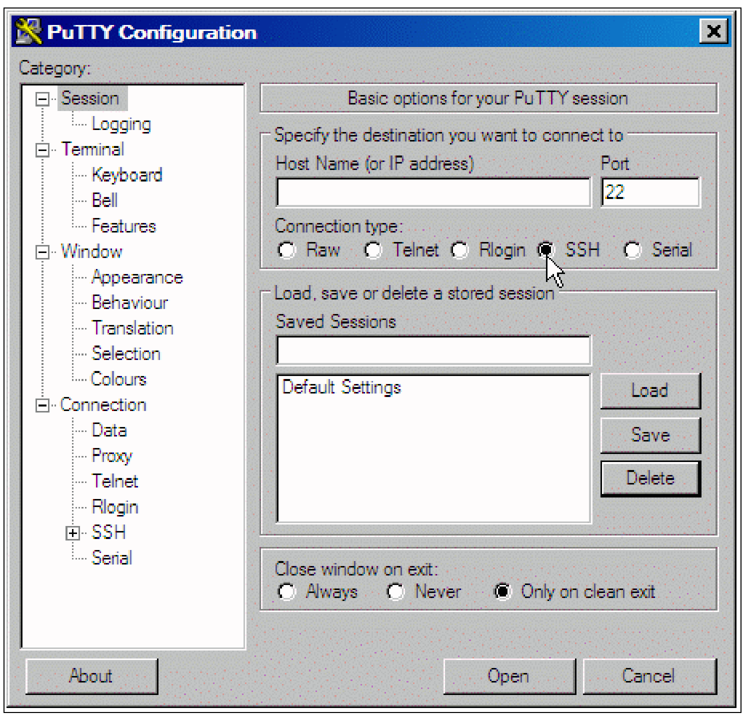


Figure 3-1 PuTTY Configuration window

- Click **Logging** in the left panel, as shown in Figure 3-2 on page 37.
  - Click the radio button **Printable output** in the *Session logging* radio group. 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 a time stamp will be in the file name.

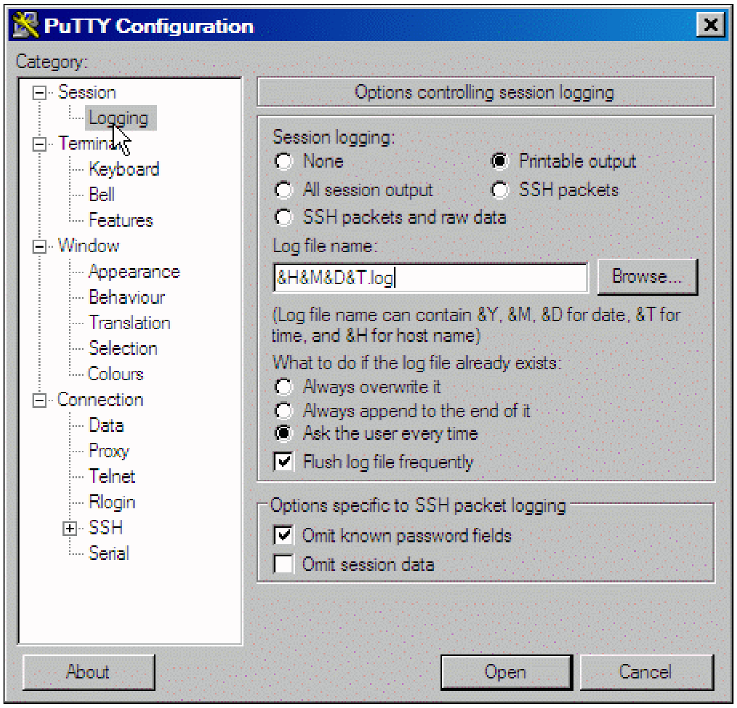


Figure 3-2 Setting logging

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

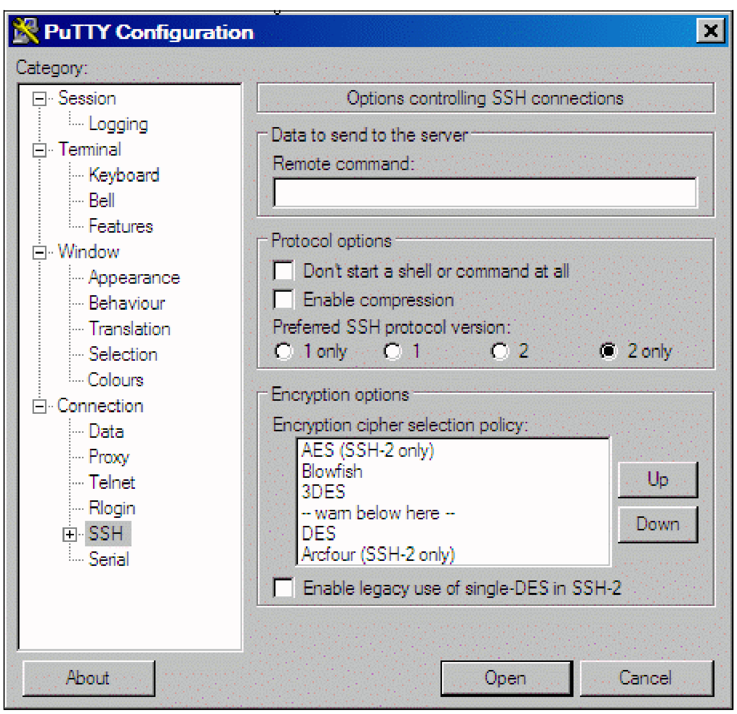


Figure 3-3 Setting SSH Protocol 2

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

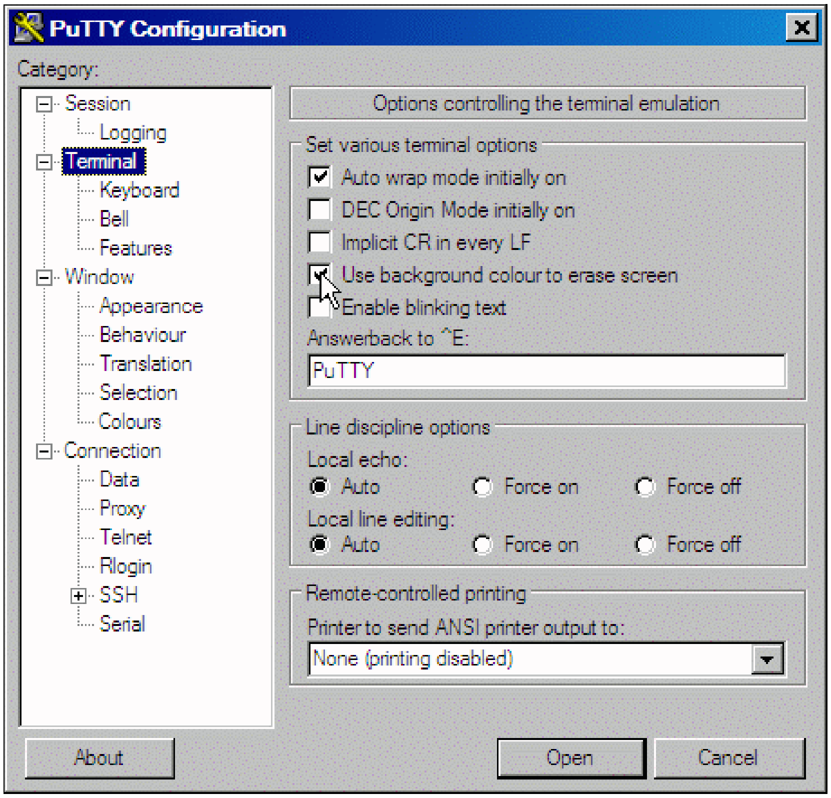


Figure 3-4 Customizing PuTTY SSH settings (Part 1 of 4)

- Click **Window** in the left pane as shown in Figure 3-5 on page 39.
- You can choose a larger window size and more lines of scrollback. In this example, 50 rows, 100 columns, and 1000 lines of scroll back are set.

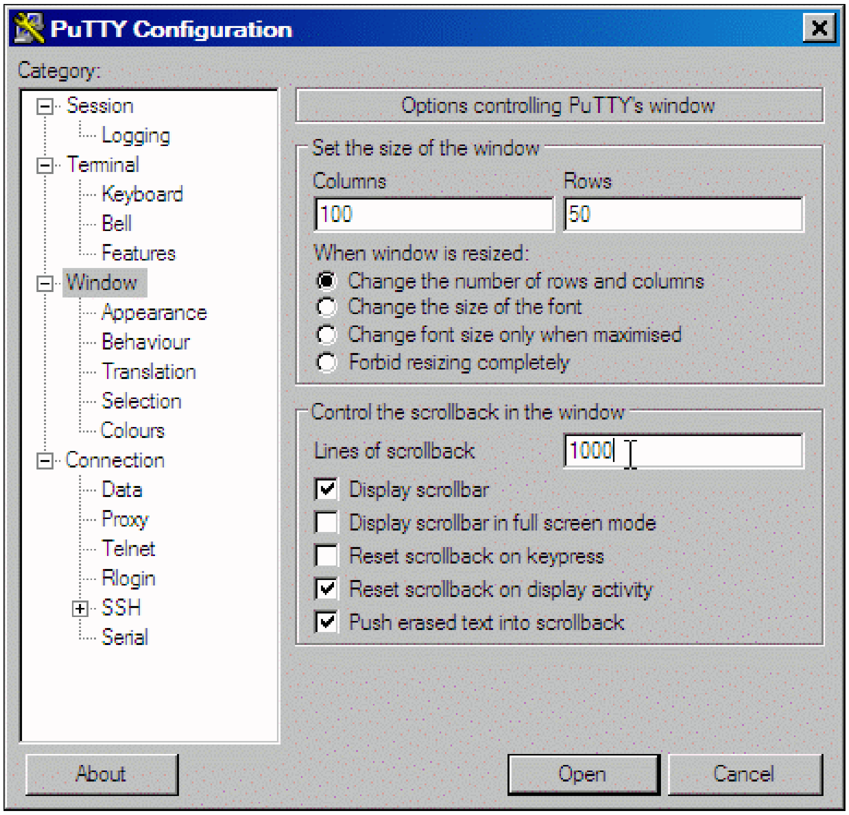


Figure 3-5 Setting window and scroll back size

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

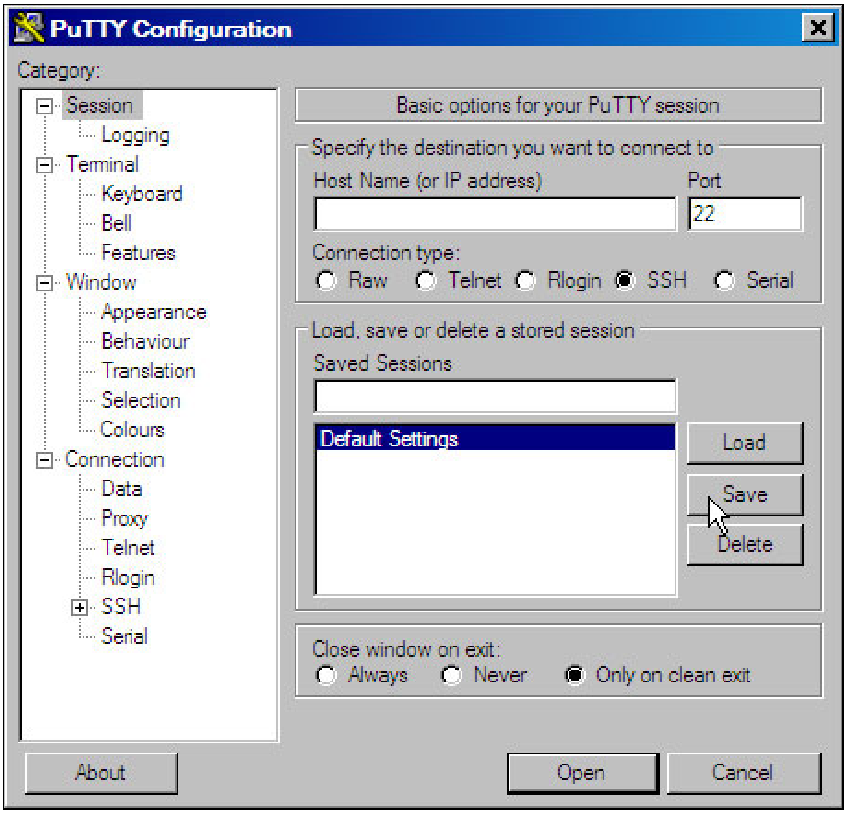


Figure 3-6 Saving new default settings

Save sessions

To save sessions, perform the following steps. In the example shown in Figure 3-7 on page 41, a session for `LINUX00` is saved.

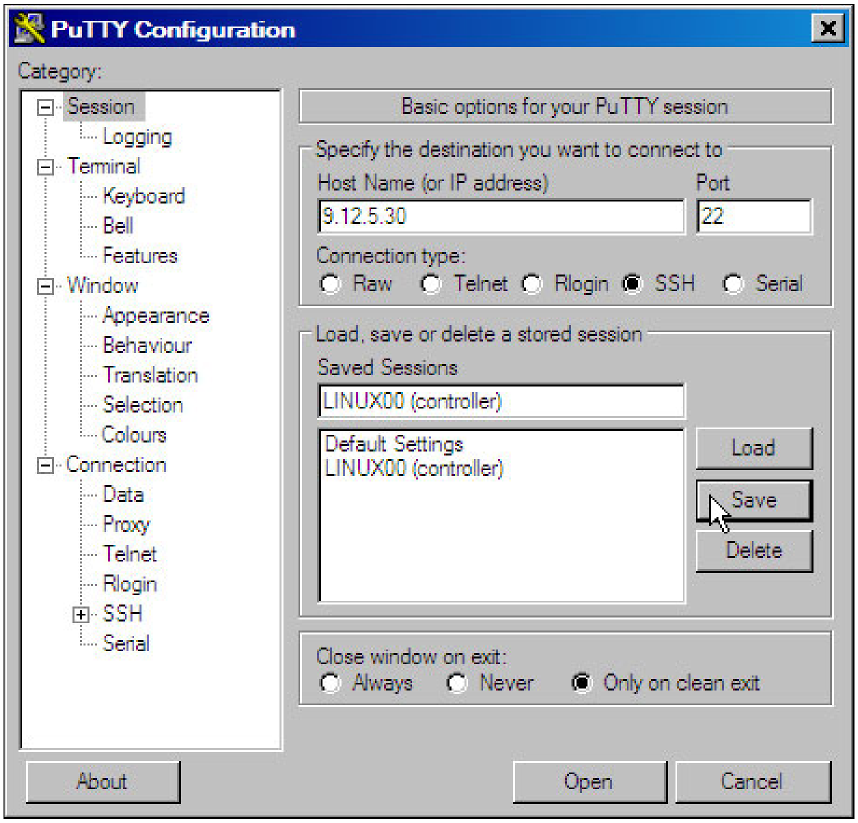


Figure 3-7 Customizing PuTTY window settings (Part 4 of 4)

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

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

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

3.2 Set up a VNC client

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

If you have a Windows desktop, the VNC client from RealVNC is a popular choice. You can purchase a full function RealVNC client, or there is a version that is available at no charge. The home page for RealVNC is at the following site:

`http://www.realvnc.com`

The download page is available at the following site:

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

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

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

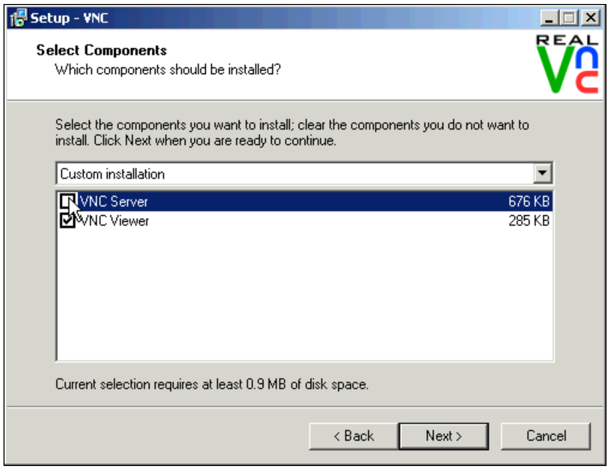


Figure 3-8 RealVNC Select Components panel

Complete the panels and the installation process should go quickly.

Attention: Although there is no specific download site for the RealVNC viewer for Windows Vista or Windows 7, instructions for both can be found at the following site:

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

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

`http://www.tightvnc.com/download.php`

The following text was found:

"TightVNC 2.0.4 supports all client and server versions of Microsoft Windows starting at Windows 2000, up to Windows 7."

3.3 3270 emulators

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

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

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 43 line screen (or more) if they can easily fit in a window given your desktop display size and resolution.
- Have the session automatically reconnect after logoff. Having a new logon window come back immediately after you log off can also save you time in the end. 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 can connect, as was described for PuTTY. Then, you can usually double-click the saved connection to quickly access a new 3270 session.

Doing a little bit of work on customizing your 3270 emulator on the front-end can save much time in the end.

3.4 Linux desktops

If you are using a Linux desktop, you should have, or at least have access to these three tools:

- An SSH client, named **ssh**. It is part of the `openssh` package.
- A VNC client, named **vncviewer**. It is part of the `tightvnc` package.
- A 3270 emulator, named **x3270**. It is part of the `x3270` package.

4

Configure an NFS/FTP server

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

— Albert Einstein

This chapter describes how to configure a PC server running Linux to be an FTP server to enable z/VM to be installed, and an NFS server to enable either RHEL 6.4 or SLES 11 SP3 (or both) to be installed. The PC server described in this chapter is shown in the lower right of Figure 1-2 on page 9.

z/VM 6.3 can be installed by using at least one of two different methods:

- From physical DVDs
- From files served by an FTP server

If you have z/VM 6.3 physical DVDs, you do not need to set up an FTP server. If you would like to avoid using physical disks and download the z/VM installation files from the Internet, you need to set up an FTP server.

After z/VM is installed, you can install Linux into a virtual machine. If you are installing Linux on System z for the first time in your organization, you will need a method of serving the installation files for at least the first installation. A PC Linux system is recommended and described in this book. It can also be a UNIX computer (Sun Solaris, Hewlett Packard HP-UX, IBM AIX®, or other).

For either RHEL or SLES, it is recommended that you install Linux onto the PC hard disk. This chapter describes the steps to do this:

- "Install Linux on a PC server" on page 46
- "Copy files to the Linux server" on page 46
- "Set up an NFS server" on page 49
- "Set up an FTP server" on page 52

4.1 Install Linux on a PC server

It is recommended that you install the same Linux distribution that you plan to install on System z onto a *distributed systems server* (for brevity, this is just called a "PC"). 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 (x86_64) and System z (s390x) architectures.

Describing in detail how to install Linux on a PC server is outside the scope of this book. However, following the steps in this section can help you.

4.1.1 Obtain RHEL for a PC server

To install RHEL 6.4, you can obtain an evaluation copy by performing the following steps:

- Access the following URL:

  `https://access.redhat.com/downloads`

- Click the **Evaluations & Demos** tab.
- Click **Red Hat Enterprise Linux 30-day Evaluation**.
- Follow the prompts to complete the form and download.
- The *Installation Guide* can be found at the following site:

  `https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Installation_Guide`

4.1.2 Obtain SLES for a PC server

To install SLES 11 SP3, you can have a licensed version on DVDs or you can choose an evaluation copy. The evaluation copy can be found at the following URL:

`http://www.suse.com/products/server/eval.html`

Follow the link for the hardware that you will install onto, and create an account to download the ISO images. When you have the ISO image, install the Linux distribution onto the PC server.

4.2 Copy files to the Linux server

By now you should have a Linux distribution running on a PC server. Take note of the host name or IP address. In this example, there was no DNS host name assigned, so an IP address will be used (in Example 4-1, we use ***9.12.5.251***):

Example 4-1 Displaying the IP address

```
# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:21:CC:B6:94:DA
          inet addr:9.12.5.251  Bcast:9.12.15.255  Mask:255.255.240.0
...
```

You now need to copy files to this server. The files fall into four categories:

- Files associated with this book
- RHEL 6.4 installation files

- SLES 11 SP3 installation files
- z/VM 6.3 installation files (if not using DVDs)

You must have the files associated with this book available and those for at least one of the two Linux distributions. If you do not have z/VM 6.3 physical DVDs and plan to install using FTP, you will also need the z/VM files.

4.2.1 Download files associated with this book

This book has files associated with it to make the task of customizing and cloning your virtual servers easier. The tar file, `SG248147.tgz`, is about 41 KB. Perform the following steps:

- Make a new directory to host the files. In this example, it is `/srv/nfs/`:

```
# mkdir /srv/nfs
# cd /srv/nfs
```

- Download the file from the web at the following site:

  http://www.vm.ibm.com/devpages/mikemac/SG248147.tgz

- It can be downloaded directly to this Linux server, or downloaded to a staging server and uploaded to the directory, `/srv/nfs/`.
- Extract the file with the following **tar** command:

```
# tar xvf SG248147.tgz
SG248147/
SG248147/disclaimer.txt
SG248147/README.txt
SG248147/vm/
SG248147/vm/lnxmaint/
SG248147/vm/lnxmaint/sample.conf-rh6
SG248147/vm/lnxmaint/sample.parm-rh6
SG248147/vm/lnxmaint/profile.exec
SG248147/vm/lnxmaint/swapgen.exec
SG248147/vm/lnxmaint/sample.parm-s11
SG248147/vm/lnxmaint/rhel64.exec
SG248147/vm/lnxmaint/sles11s3.exec
SG248147/vm/maint/
SG248147/vm/maint/ssicmd.exec
SG248147/vm/maint/callsm1.exec
SG248147/vm/maint/cpformat.exec
SG248147/rhel64/
SG248147/rhel64/clone-1.0-11.s390x.rpm
SG248147/sles11sp3/
SG248147/sles11sp3/linux5.xml
SG248147/sles11sp3/jeos.tgz
SG248147/sles11sp3/boot.clone
SG248147/sles11sp3/clone.sh
```

- It is recommended that you review the file `README.txt`.

You now have access to the files associated with this book.

4.2.2 Obtain RHEL 6.4 for System z

If you plan to install RHEL 6.4, you can request a 90-day evaluation copy at no charge by performing the following steps:

- Access the following URL:

  http://www.redhat.com/z

- Click the link **Evaluation software download** on the left.
- Complete the online form. If you do not have a Red Hat login, you will need to create one by clicking the **SIGN UP** button. Otherwise, enter your Red Hat login and password, then click **Log In** to continue.
- After completing the form, you will automatically receive an email with instructions on how to access the Red Hat Network (RHN), where you can download the installation discs that are found at the following URL:

  https://rhn.redhat.com

- Download the ISO image file. You can download it directly to the directory `/srv/nfs/`, or you can stage it on a workstation and upload it to that directory.

Verify the RHEL 6.4 ISO image

An important early step is to verify the integrity of the 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, there was an error somewhere in the copying process. The following functions can help you evaluate your checksum values:

- The **sha1sum** command allows you to compare checksum files. The checksum value for RHEL 6.4 DVD for the s390x architecture is as follows:

```
# cat SHA1SUM
a3bf31bd7223b9cceddfce03963b7244a0784d87  rhel-server-6.4-s390x-dvd.iso
```

- If you did not also download an `SHA1SUM` file, you can copy and paste the line above into a file to create one.
- Run the **sha1sum -c** command against the `SHA1SUM` file to verify the contents of the ISO image:

```
# sha1sum -c SHA1SUM
rhel-server-6.4-s390x-dvd.iso: OK
```

If the output of the command does not reply with `OK`, you must get the correct ISO file before proceeding.

4.2.3 Obtain SLES 11 SP3 for System z

If you plan to install SLES 11 SP3, perform the following steps to obtain an evaluation copy:

1. Access the following URL:

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

2. Click the **System z** link.
3. Click **proceed to download** in the upper right part of the page.
4. Complete the online form to complete the process.

Verify the SLES 11 SP3 ISO images

Verify the integrity of the ISO images. This is done by using a file of checksum values and ISO file names. The checksums were calculated from the contents of the DVD. After downloading or **dd**ing the ISO images, they can be checked using a program called **checkmedia**:

```
# checkmedia SLES-11-SP3-DVD-s390x-GMC-DVD1.iso
    app: SLES-11-SP3-DVD-s390x-Build0664-Media1
  media: DVD1
   size: 3227906 kB
 hybrid: 3228672 kB
    pad: 300 kB
  check: 100%
  check: md5sum ok
    md5: 5ee12747107ee1fa3051eb7ee9f42b54
```

Run this program for all of the downloaded ISO images. Any ISO images that do not report `OK` must be downloaded or copied again.

4.2.4 Obtain z/VM 6.3

If you have DVDs to install z/VM, you do not need to obtain the files electronically. If you choose to obtain z/VM 6.3 from the Internet, see section 5.1, “Obtain z/VM through electronic download” on page 58.

4.3 Set up an NFS server

In this section, we provide the following steps to set up an NFS server for the first Linux on System z installation:

- “Mount the Linux distribution ISO image for RHEL 6.4” on page 49
- “Turn on the NFS server” on page 51
- “Test the NFS server” on page 51

4.3.1 Mount the Linux distribution ISO image for RHEL 6.4

If you are using RHEL 6.4, a single ISO image needs to be mounted. To do so, perform the following steps:

1. Upload the RHEL 6.4 ISO image file and verify. In this example it is uploaded to `/srv/nfs/`:

   ```
   # cd /srv/nfs
   # ls -l rhel*
   -rwxr-xr-x. 1 root root 2916585472 Jun 25 09:50 rhel-server-6.4-s390x-dvd.iso
   ```

2. Create a directory over which the ISO image will be mounted. In this example, it is `/srv/nfs/rhel64/`:

   ```
   # mkdir rhel64
   ```

3. Mount the ISO image by using a loopback device with the **mount -o loop** command:

   ```
   # mount -o loop rhel-server-6.4-s390x-dvd.iso rhel64
   ```

4. List the contents of the mounted ISO image:

   ```
   # ls /srv/nfs/rhel64
   boot.cat     Packages                 RELEASE-NOTES-mr-IN.html
   EULA         README                   RELEASE-NOTES-or-IN.html
   ...
   ```

You should now have the contents of the RHEL 6.4 DVD accessible in the `/srv/nfs/rhel64/` directory.

4.3.2 Install and configure the NFS server

An NFS server needs to be installed and configured. In this section, we provide instructions on how to do this.

On RHEL 6.4

If you are using RHEL 6.4, run the following command from the command prompt:

```
# yum install nfs-utils
...
```

The NFS server is now installed.

On SLES 11 SP3

If you are on SLES 11 SP3, perform the following step from a command prompt:

- Use the **zypper** installation command to install the `nfs-kernel-server` RPM:

```
# zypper in nfs-kernel-server
Loading repository data...
Reading installed packages...
Resolving package dependencies...

The following NEW package is going to be installed:
  nfs-kernel-server

1 new package to install.
Overall download size: 111.0 KiB. After the operation, additional 247.0 KiB
will be used.
Continue? [y/n/?] (y):
Retrieving package nfs-kernel-server-1.2.3-18.29.1.x86_64 (1/1), 111.0 KiB (247.0 KiB
unpacked)
Installing: nfs-kernel-server-1.2.3-18.29.1 [done]
```

You now have an NFS server installed.

4.3.3 Mount the Linux distribution ISO images for SLES 11 SP3

If you are using SLES 11 SP3, two ISO images need to be mounted. To do so, perform the following steps:

1. Create a directory over which the ISO image will be mounted. In this example, it is `/srv/nfs/s11s3/`:

```
# cd /srv/nfs
# mkdir s11s3
```

2. Upload the two ISO image files and verify. In this example, the /srv/nfs/s11s3/ directory is used:

```
# ls -l SLE*
-rwxr-xr-x. 1 root root 2233427968 Jun 25 11:31 SLE-11-SP3-SDK-DVD-s390x-GMC-DVD1.iso
-rwxr-xr-x. 1 root root 3301965824 Jun 25 09:41 SLES-11-SP3-DVD-s390x-GMC-DVD1.iso
```

3. Mount the first ISO image by using a loopback device with the **mount -o loop** command (the second ISO image does not need to be loopback-mounted):

```
# mount -o loop SLES-11-SP3-DVD-s390x-GMC-DVD1.iso s11s3
```

4. List the contents of the mounted ISO image:

```
# ls s11s3
ARCHIVES.gz  COPYRIGHT.de                        license.tar.gz
boot         directory.yast                      ls-lR.gz
ChangeLog    docu                                media.1
content      gpg-pubkey-307e3d54-4be01a65.asc  NEWS
content.asc  gpg-pubkey-39db7c82-510a966b.asc  pubring.gpg
content.key  gpg-pubkey-3d25d3d9-36e12d04.asc  README
control.xml  gpg-pubkey-50a3dd1c-50f35137.asc  suse
COPYING      gpg-pubkey-9c800aca-4be01999.asc  suse.ins
COPYING.de   gpg-pubkey-b37b98a9-4be01a1a.asc
COPYRIGHT    INDEX.gz
```

The installation files for SLES 11 SP3 should now be ready to be NFS-exported.

4.3.4 Configure the files to be exported

The configuration for the NFS server is stored in the `/etc/exports` file. Execute the commands that are shown in Example 4-2 and edit the **`exports`** file as shown in the example to set up an NFS server for the RHEL installation and make the contents of the files associated with this book available.

Example 4-2 Export configuration

```
# cd /etc
# cp exports exports.orig
# vi exports
/srv/nfs/rhel64       *(ro,root_squash,sync)
/srv/nfs/s113         *(ro,root_squash,sync)
/srv/nfs/SG248147     *(ro,root_squash,sync)
```

4.3.5 Turn on the NFS server

The NFS server should now be configured. To start it, perform the following steps:

1. Set the NFS server to start at boot time with the following **`chkconfig`** command:

```
# chkconfig nfs on
```

2. Start the NFS service for this session with the following **`service`** command:

```
# service nfs start
Starting NFS server:                                [ OK ]
```

The NFS server will now always be started.

4.3.6 Test the NFS server

NFS should now be running on the system. In order to test it, list the exported NFS directories with the **`showmount -e`** command:

```
# showmount -e
Export list for localhost.localdomain:
/srv/nfs/SG248147 *
/srv/nfs/s11s3    *
/srv/nfs/rhel64   *
```

The output of this command shows that the NFS service is up and the file systems are available for mount.

If you have a problem accessing the services remotely, ensure that the firewall is either switched off, or configured to permit those services.

4.4 Set up an FTP server

If an FTP server is to be installed and configured, the **vsftpd** FTP server is recommended. This section shows how to configure it as an anonymous FTP server.

4.4.1 Install and configure the FTP server on RHEL

If you installed RHEL on to the PC server, use the following steps to install and configure the FTP server:

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

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

   No output shows that it is not installed.

2. If it is not installed, use the **yum -y** command to install the package:

   ```
   # yum -y install vsftpd
   Loaded plugins: rhnplugin
   This system is not registered with RHN.
   ...
   Installed:
   vsftpd.s390x 0:2.2.2-6.el6
   ```

3. Make a backup of the configuration file, /etc/vsftpd/vsftpd.conf:

   ```
   # cd /etc/vsftpd
   # cp vsftpd.conf vsftpd.conf.orig
   ```

4. Ensure that **anonymous_enable=YES** is uncommented at the /etc/vsftpd/vsftpd.conf configuration file:

   ```
   # vi vsftpd.conf
   ...
   # Example config file /etc/vsftpd/vsftpd.conf
   # Allow anonymous FTP? (Beware - allowed by default if you comment this out).
   anonymous_enable=YES
   ...
   ```

5. Set up SELinux to comply with this configuration by enabling the boolean that allows the FTP server to share the ISO image bind mounted:

   ```
   # setsebool -P allow_ftpd_full_access on
   ```

6. Change the home directory to /srv/ftp/ instead of the default /var/ftp/ by using the **usermod -d** command:

   ```
   # usermod -d /srv/ftp ftp
   ```

7. Set the **vsftpd** service to start at boot time with the **chkconfig** command:

   ```
   # chkconfig vsftpd on
   ```

8. Turn on the FTP server for this session with the **service** command:

   ```
   # service vsftpd start
   Starting vsftpd for vsftpd: [ OK ]
   ```

An anonymous FTP server will now be running.

4.4.2 Install and configure the FTP server on SLES

A normal installation of SLES11 SP3 does not install an FTP or NFS server. The **vsftpd** server is recommended as the FTP server. The FTP client that is built into the Hardware Management Console (HMC) requires that the FTP server prompt for a password. This is the default with **vsftpd**.

If you installed SLES on the PC server, perform the following steps to install and configure the FTP server:

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

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

   No output shows that it is not installed.

2. Use the **zypper** installation command to install the package:

   ```
   # zypper in vsftpd
   Loading repository data...
   Reading installed packages...
   Resolving package dependencies...

   The following NEW package is going to be installed:
     vsftpd

   1 new package to install.
   Overall download size: 126.0 KiB. After the operation, additional 290.0 KiB
   will be used.
   Continue? [y/n/?] (y):
   Retrieving package vsftpd-2.0.7-4.25.1.x86_64 (1/1), 126.0 KiB (290.0 KiB unpacked)
   Installing: vsftpd-2.0.7-4.25.1 [done]
   ```

3. The default is to operate as an anonymous FTP server so no changes are needed in the configuration file `/etc/vsftpd.conf`.

4. The default home directory of the FTP server should be `/srv/ftp/`. Verify this by using the following two commands:

   ```
   # cd ~ftp
   # pwd
   /srv/ftp
   ```

5. Create a new directory for FTP files:

   ```
   # mkdir -p /srv/ftp/sles11sp3/
   ```

The anonymous FTP server will now be configured.

4.4.3 Test the anonymous FTP server

Test that the FTP server can be accessed as anonymous with the **ftp localhost** command and try to log in as `anonymous`:

```
# ftp localhost
Connected to localhost (127.0.0.1).
220 (vsFTPd 2.2.2)
Name (localhost:root): anonymous
331 Please specify the password.
Password: <anyPassword>
```

```
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> quit
```

This shows that the anonymous FTP server is working.

4.5 Set up an SLES 11 SP3 installation server

There are different possible methods of making the SLES 11 SP3 distribution available:

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

A combination of the last two options is described in this section. The ISO images are made available with NFS, and the loop mount is used to make an FTP repository of DVD 1 available.

4.5.1 Install FTP server packages

In this section, we describe the steps needed to install the FTP server packages.

Obtain the ISO images

- When starting from a softcopy of the DVD:

  You should have the DVD ISO images for SLES 11 SP3 (at the time this book was written, we used *Release Candidate 2*). If you are starting with a softcopy of the DVD1 ISO image, *copy it* to the `/srv/nfs/s11s3/` directory now.

- When starting from physical DVDs:

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

  Perform these steps only if you are starting with an SLES 11 SP3 distribution for System z on physical DVD discs:

  a. Put the first DVD into the CD/DVD drive. The disc should be automatically mounted over the directory `/dev/cdrom/` (if you cannot find it there, try `/dev/dvdrom/`, `/dev/sr0`, or `/dev/dvdrw/`).

  b. Make a directory to store the ISO image and change it by using the following commands:

     # **mkdir -p /srv/nfs/s11s3**

     # **cd /srv/nfs/s11s3**

  c. Use the **dd** command with the **if** (input file) and **of** (output file) parameters to copy the contents of the disc to an ISO image:

     # **dd if=/dev/cdrom of=SLES-11-SP3-DVD-s390x-GM-DVD1.iso**

  d. Unmount the DVD disc:

     # **umount /mnt/cdrom**

  e. Repeat the process for the SDK disc by using the following target name: `SLE-11-SP3-SDK-DVD-s390x-GMC-DVD1.iso`

You should now have ISO images of the physical DVDs.

Set up the FTP tree

Make the contents of the ISO image available to FTP. This is done by loop mounting the ISO image to a directory below `/srv/ftp/`:

1. Create the directory `/srv/ftp/SLES-11-SP3-RC2/`:

```
# mkdir -p # /srv/ftp/SLES-11-SP3-RC2/
```

2. Add an entry to the `/etc/fstab` file to automatically create a loop mount. To accomplish this, add the following line:

```
# vi /etc/fstab
...
/srv/nfs/s11s3/SLES-11-SP3-DVD-s390x-RC2-DVD1.iso /srv/ftp/SLES-11-SP3-RC2 iso9660 loop,ro
```

3. Run the **mount** command to activate the new entry:

```
# mount -a
```

4. Check if the mount was successful:

```
# ls /srv/ftp/SLES-11-SP3-RC2
ARCHIVES.gz   directory.yast
COPYING       docu
...
```

This shows that the ISO image was successfully mounted.

4.5.2 Create the exports for the NFS directory

The configuration for the NFS server is stored in the file `/etc/exports`. The directories `/srv/nfs/SG248147/`, and `/srv/nfs/s11s3/` are exported using the following commands and by editing the **exports** file:

```
# cd /etc
# cp exports exports.orig
# vi exports
/srv/nfs/s11s3          *(ro,root_squash,sync,no_subtree_check)
/srv/nfs/SG248147       *(ro,root_squash,sync,no_subtree_check)
```

4.5.3 Update the system services configuration

At this point, the FTP and NFS servers have been configured but not started. To start them, perform the following steps:

1. Enable the FTP and NFS servers started at boot time:

```
# chkconfig vsftpd on
# chkconfig nfsserver on
```

2. Start the services for this session:

```
# rcvsftpd start
Starting vsftpd                                                      done
# rcnfsserver start
Starting kernel based NFS server: idmapd mountd statd nfsd sm-notify done
```

4.5.4 Test the FTP and NFS servers

The FTP and NFS servers are now in service to the system. To test, perform the following steps:

1. Run **ftp** to access the local host:

```
# ftp localhost
Connected to localhost.
220 (vsFTPd 2.0.7)
Name (localhost:root): ftp
331 Please specify the password.
Password:
ftp> ls
drwxr-xr-x    5 0        0              82 Jun 14 17:17 SG248147
dr-xr-xr-x    6 0        0            4096 Jun 06 19:53 SLES-11-SP3-RC2
```

2. Access the exported NFS directories locally:

```
# showmount -e
Export list for instserver:
/srv/nfs/SG248147 *
/srv/nfs/s11s3    *
# mount localhost:/srv/nfs/s11s3 /mnt
# ls /mnt
SLES-11-SP3-DVD-s390x-GMC-DVD1.iso
```

This shows that both services are up and running correctly. If you have a problem remotely accessing the services, make sure that the firewall is either switched off, or configured to permit those services.

5

Install a z/VM SSI cluster

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

— Albert Einstein

In this chapter, we describe how to install a z/VM single system image (SSI) cluster.

z/VM 6.3 can be installed from a DVD disk or from an FTP server. Installing from an FTP server is described in this section; installing from a DVD is also mentioned. This chapter consists of the following sections that should be completed in their entirety because the chapters that follow rely on these changes:

- "Obtain z/VM through electronic download" on page 58
- "Configure an FTP server for z/VM installation" on page 59
- "Install z/VM from DVD or FTP server" on page 60
- "Configure TCP/IP" on page 71
- "Configure the XEDIT PROFILE" on page 74
- "Customize the SYSTEM CONFIG file" on page 76
- "Configure additional network resources" on page 78
- "Add page and perm volumes" on page 82
- "Configure AUTOLOG1's PROFILE EXEC" on page 88
- "Create LNXMAINT for common files" on page 90
- "Create identity LNXADMIN for Linux administration" on page 96

In addition, there are optional sections:

- "z/VM security issues" on page 99
- "Back up and restore your z/VM system" on page 100

Important: These are not official instructions about how to install z/VM 6.3. The *z/VM Installation Guide, version 6 release 3* can be found under the "Installation, Migration, and Service" section in the z/VM 6.3 Information Center, which is at the following link:

`http://pic.dhe.ibm.com/infocenter/zvm/v6r3/index.jsp`

This chapter describes installing z/VM 6.3 from an FTP server onto a two-node SSI cluster residing on DASD. It also addresses installing from DVDs. If you are installing onto SCSI disks or using different options, you should use the official z/VM documentation.

5.1 Obtain z/VM through electronic download

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

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

To order z/VM, perform the following steps:

1. Go to the z/VM service page:

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

2. Click the link **IBM Shopz** in the section *IBM Support Portals*. Also, this can be reached directly from the following URL:

   `http://www.ibm.com/software/ShopzSeries`

3. Sign in by clicking the link **Sign in for registered users** in the upper right. Note that there is a separate line for IBM employees to use their intranet credentials.
4. Click the link **create new software orders** for service or products.
5. On *My orders*, click the radio button **z/VM Products** and choose **VM SDO version 6** in the drop-down menu to the right. Click **Continue**.
6. On *Step 1 of 7*, accept the *Order Name* and click **Continue**.
7. On *Step 2 of 7*, select a hardware system on which you plan to run z/VM from the list of hardware systems for your customer number, and click **Continue**.
8. On *Step 3 of 7*, for the *Group*, select **VM - VM Base Product**. Select your language, and for the *Filter*, select **Show all products**. Then click **Show catalog**. A submenu appears.
9. Select **z/VM V6 3390 System DDR** and click **Continue**.
10. On *Step 4 of 7*, verify the order and click **Continue**.
11. On *Step 5 of 7*, verify the entitlements and click **Continue**.
12. On *Step 6 of 7*, for the *Preferred media*, select **Internet** and click **Continue**.
13. On *Step 7 of 7*, review and click **Submit**.

    It might take some time for the order to be prepared. In our example, the email stating that the order was ready for download was received after about 4 hours. When you receive the email, it contains the URL for downloading your order. Use a browser to go to that URL.

    There will be links to investigate, as shown in Figure 5-1 on page 59. It has the following five sections:

    – *Order Packing List*: Contains the list of available products and manuals

    – *Installation Instructions*: Clicking **View now** takes you to a web page:

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

    – *Product Publications*: Allows you to access different z/VM publications that are related to installation

    – *Additional Publications*: Allows you to download a z/VM SDO document (four pages)

    – *VM product material: This is the most important section* because it is where you go to download z/VM product installation files. In the example used in this book, the link

Download to your workstation using IBM Download Director was clicked as shown in the figure.

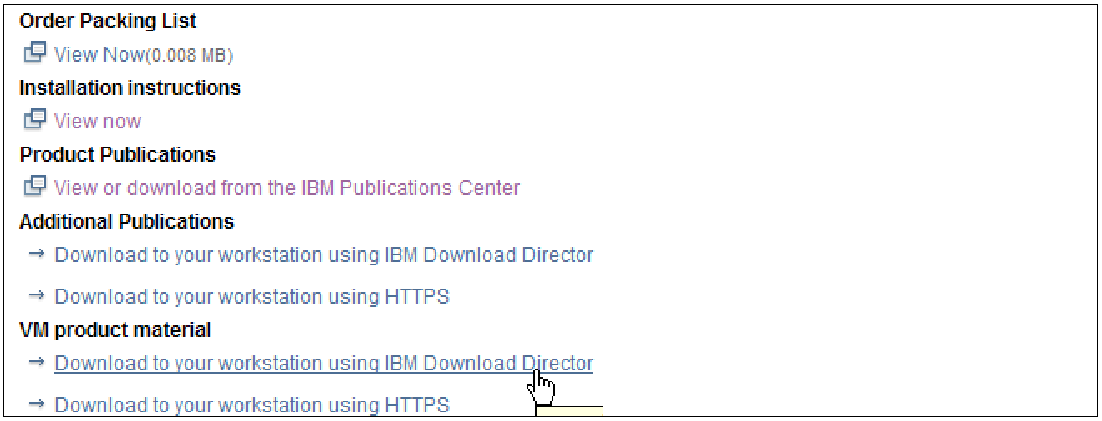


Figure 5-1 Web page for downloading z/VM electronically

14. Clicking this link displays the window that is shown in Figure 5-2. The second and third check boxes were selected as z/VM 6.3 is being installed onto 3390 DASD. The 1.3 GB of data was downloaded relatively quickly due to multiple connections being opened by using *IBM Download Director*.

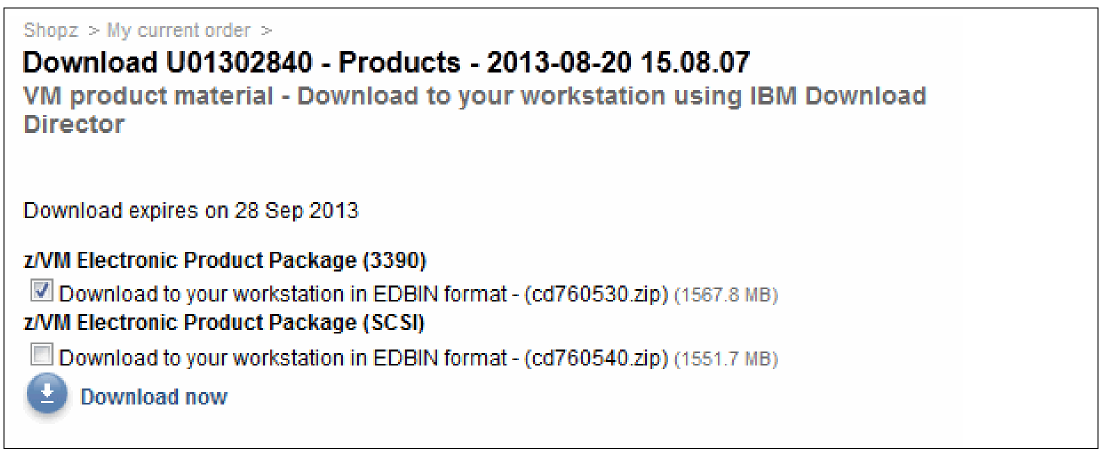


Figure 5-2 Choosing two files to be downloaded

15. The z/VM installation code should now be staged or ready for the FTP server to be set up. In our example, the file is staged on an MS Windows workstation. The files are displayed from a DOS prompt:

```
C:\zvm63> dir
...
08/29/2013  12:33 PM      1,567,838,350 cd760530.zip
```

You now have the z/VM installation compressed file downloaded. You can now set up an FTP server.

5.2 Configure an FTP server for z/VM installation

This section assumes that you have access to the z/VM 6.3 installation code in electronic format. Ordering it through *IBM Shopz* is briefly described in section 5.1, “Obtain z/VM through electronic download” on page 58. If you have completed that section, you may have one compressed file containing the z/VM product install files staged on an intermediate workstation, or you may be ready to download it from the Internet.

5.2.1 Prepare the z/VM product installation files

The compressed file contains z/VM product DVD. The content of this file must be copied to the directory of the FTP server. To accomplish this, perform the following steps:

1. Start an SSH session to the PC NFS server set up in Chapter 4, “Configure an NFS/FTP server” on page 45.
2. Create a directory where the files will be stored. In this example, it is `/srv/ftp/zvm63/`:
   ```
   # mkdir -p /srv/ftp/zvm63
   ```
3. Set the group ownership of this directory, **ftp**. This will allow the FTP daemon, which runs as the user `ftp`, to change directory into it:
   ```
   # chgrp ftp /srv/ftp/zvm63
   ```
4. Either upload the two z/VM installation compressed files from the intermediate workstation, or download them directly from the Internet. The following example shows copying them from an intermediate workstation MS Windows DOS session to the FTP server at the IP address (***9.12.5.251*** in this example) to the directory, using the add-on **pscp** command (Putty scp):
   ```
   C:\zvm63>pscp *.zip root@9.12.5.251:/srv/ftp/zvm63
   ...
   ```
5. List the newly copied files:
   ```
   # cd /srv/ftp/zvm63
   # ls -l
   ```
6. Decompress the file using the **unzip** command. This creates the directory `cpdvd/`:
   ```
   # unzip CD749500.zip
   ```

You should now have all the z/VM product installation files ready.

> **Note:** In the past, z/VM came with two compressed files. The first contained the GA level of z/VM and the second contained the first recommended service upgrade (RSU). z/VM V6R3 comes as just one compressed file, which already contains the RSU file and this RSU is applied automatically during the z/VM installation. After z/VM is installed, it is strongly recommended to check the RSU of the installed system and compare it to the latest available RSU.

5.3 Install z/VM from DVD or FTP server

In this section, we assume a first-level SSI installation of z/VM from DVD or FTP server onto 3390 DASD. If you are planning a non-SSI installation, see Chapter 7, “Install a z/VM non-SSI LPAR” on page 119.

If you are not familiar with the HMC and z/VM, you might want to use the official z/VM manual: *z/VM 6.3 Installation Guide*, GC24-6246, which is available at the following link:

http://publibz.boulder.ibm.com/epubs/pdf/hcsk2c20.pdf

If you are installing z/VM at the *second level* (z/VM under z/VM) or onto FCP/SCSI disk, you will want to use this z/VM manual because the sections that follow do not address these options.

5.3.1 Start the z/VM installation

The figures in this book show the HMC *tree view*, not the *classic view*. An example of the main menu in tree view mode is shown in Figure 5-3.

To change between the two HMC views, select **Tasks Index** on the left, then **User Settings** on the right then the **UI Style** tab.

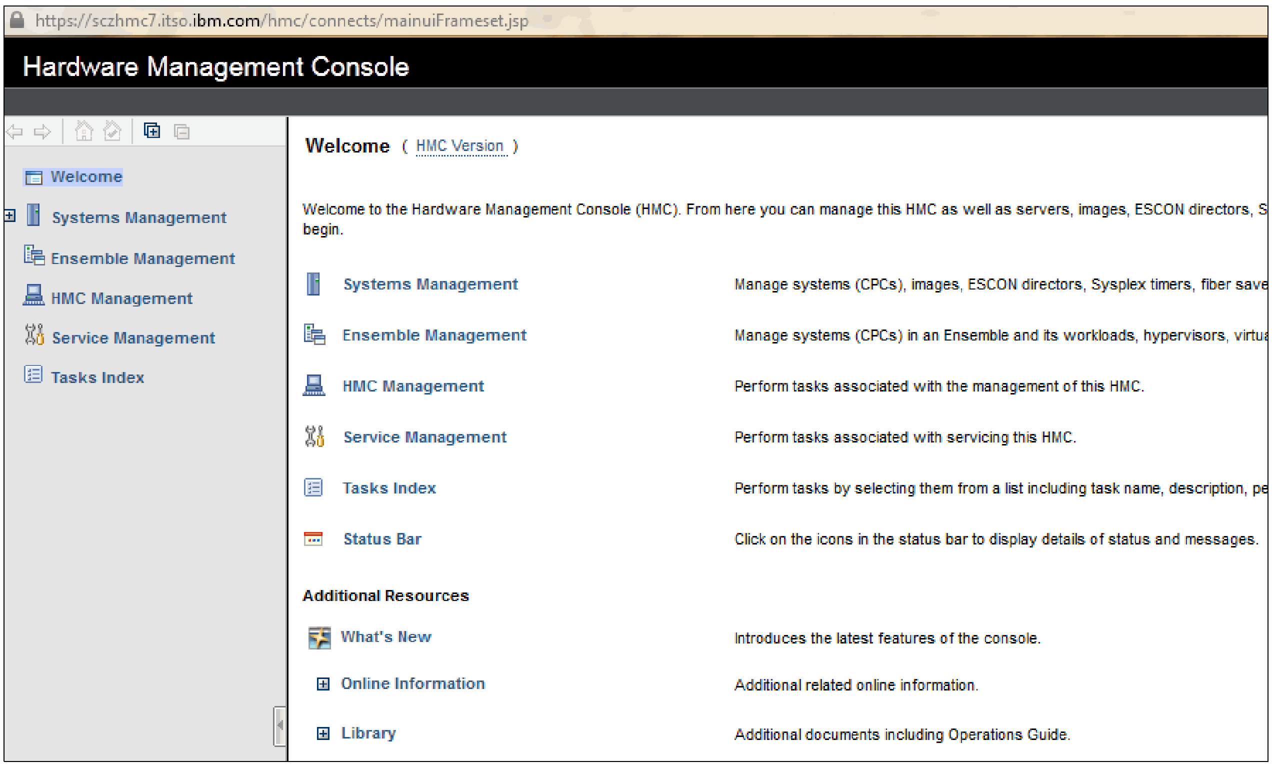


Figure 5-3 HMC tree view

To begin the z/VM 6.3 installation, perform the following steps:

1. Log on to the Hardware Management Console. You will need either physical access to the console or a URL for the web interface. You will need a user ID and password. Assuming the view is tree mode, you should see a window similar to Figure 5-3.
2. Expand the **Systems Management** link on the main pane by clicking the “+”.
3. Expand **Systems** and expand the view of the CECs where the LPARs that you will be installing onto are.
4. Select the LPAR that will be the first member of the z/VM 6.3 SSI cluster. *Be sure* that you have the correct LPAR selected. If you are not completely sure, check with someone who is. The first LPAR onto which the SSI was installed into in this example (**`A02`**) is shown in Figure 5-4 on page 62. Note the check box to the left of it.

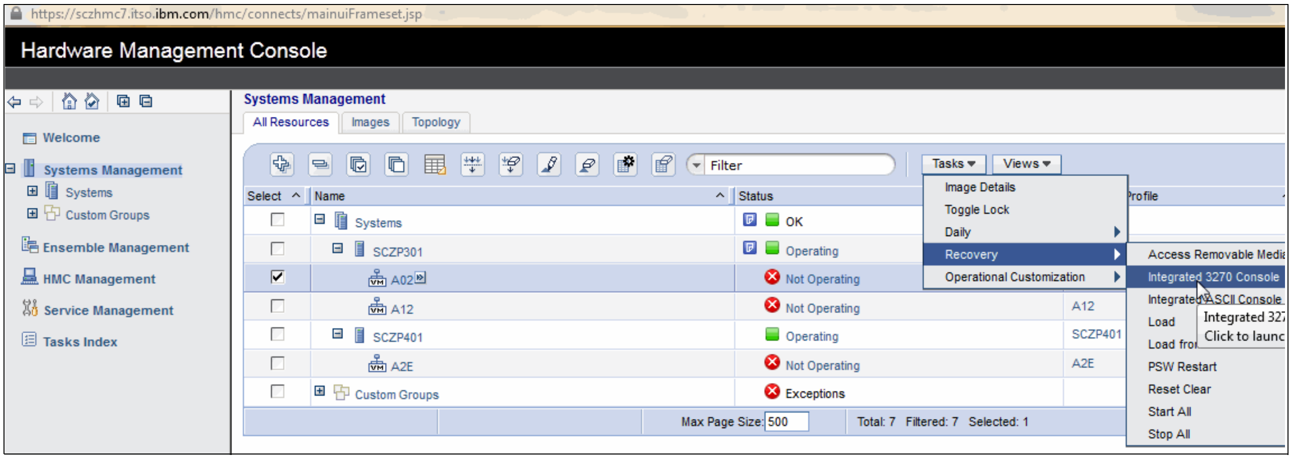


Figure 5-4 HMC with the Systems selection expanded, and Integrated 3270 Console menu item

5. Open an **Integrated 3270 Console** by clicking the **Tasks** drop-down menu in the upper right, then the **Recovery** slide-right menu. This menu sequence is shown in Figure 5-4. A new window entitled *Integrated 3270 Console* should open.
6. Begin the installation process by selecting **Load from Removable Media or Server** from the same **Recovery** submenu.
7. The *Load from Removable Media or Server* window should open, as shown in Figure 5-5 on page 63. Do the following steps:
   a. Click the radio button **FTP Source**.
   b. Enter the host name or IP address that the FTP server is running on in the *Host computer* field.
   c. Enter the FTP *User ID* and *Password*.
   d. In the *File location* field, enter the directory where the `630vm.ins` file is. In this example, it is **/srv/ftp/zvm63/cpdvd**.
   e. Click **OK**.

> **If installing from DVDs:** If you are installing from DVDs, the first disc must be in the HMC DVD drive. Click just the Hardware Management Console CD/DVD-ROM radio button. The *FTP Source* section is not used.

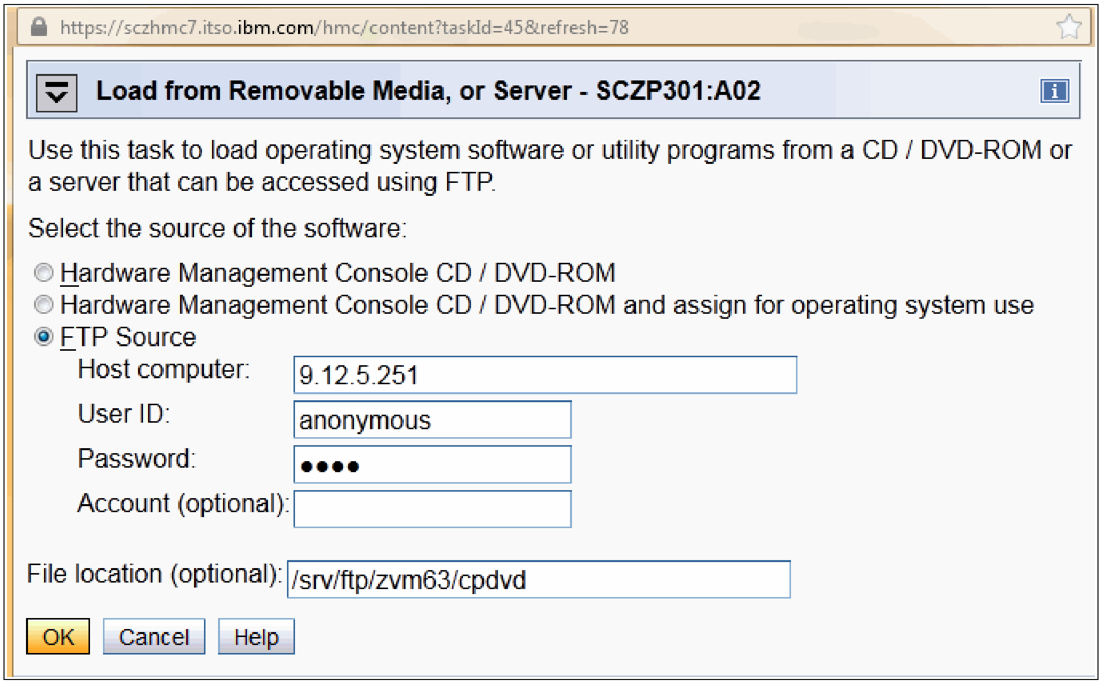


Figure 5-5 Load from Removable Media or Server panel

8. Load the RAMDISK:
   a. From the *Load from Removable Media or Server* panel, the directory containing the file **630VM.INS** should be selected. Click **OK**.
   b. From the *Confirm the action* window, click **Yes**.
   c. You should see the *Disruptive Task Confirmation: Load from CD-ROM, DVD or Server Progress* window. You *might* be prompted for the password depending on how your HMC is configured.
   d. You should see the *Load from Removable media or Server Progress* window. When you see the message `Completed successfully`. Click **OK** to close. This should normally take less than a minute.

You now have an in-memory z/VM 6.3 system running.

5.3.2 Copy a vanilla z/VM system to DASD

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

1. Move to the *Integrated 3270 Console* window. The RAMdisk should IPL and you should see z/VM boot as shown in Figure 5-6 on page 64. If the *Integrated 3270 Console* window is still blank, be patient; it can take a minute or two to initialize.

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

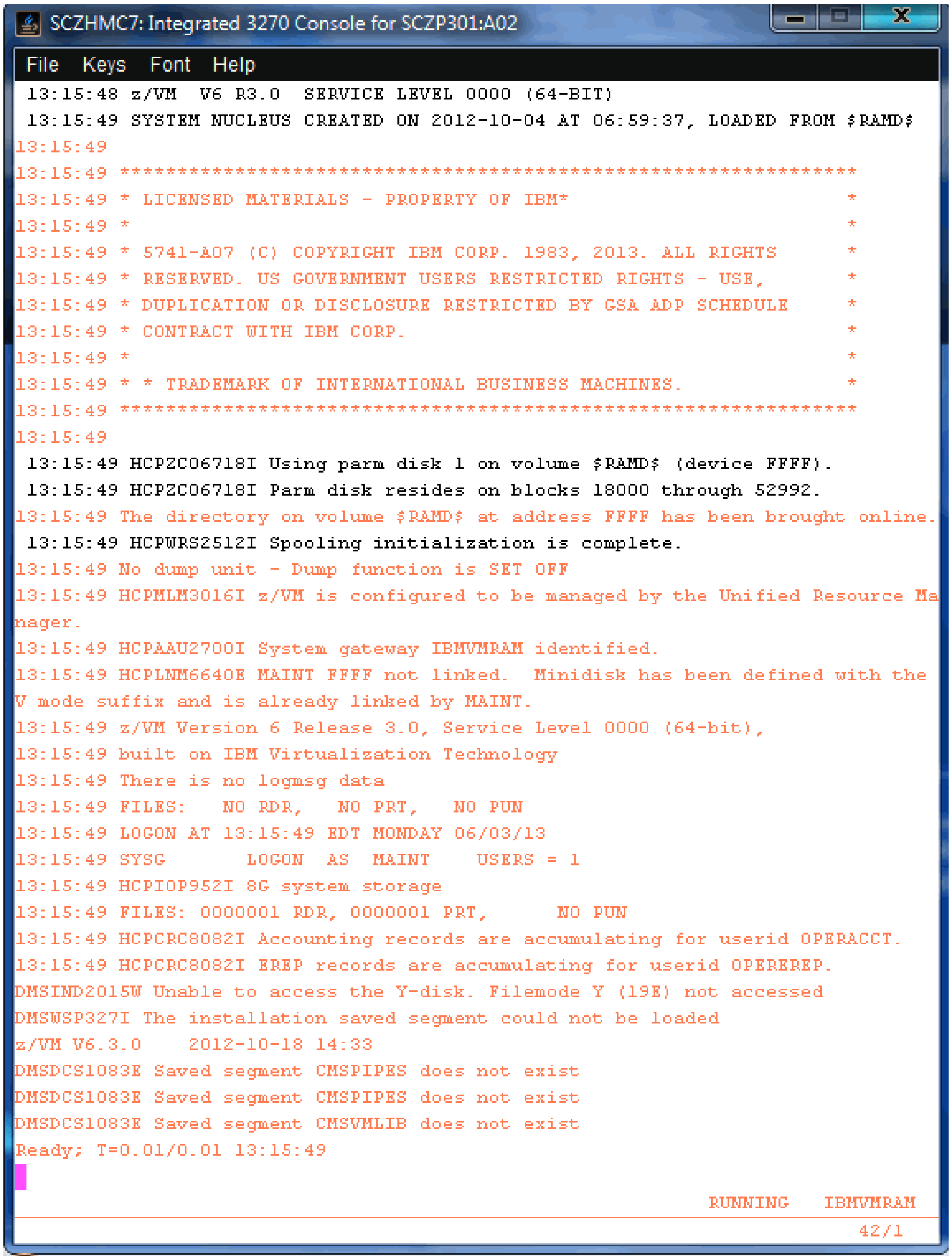


Figure 5-6 First z/VM 6.3 installation window

Run DVDPRIME

1. Run the **`DVDPRIME`** command. The format is **`dvdprime`** ***`dasdtype`*** **`(`*****`source`***. In this example, the ***`dasdtype`*** is **`3390`** and the ***`source`*** is **`server`** - for FTP server:

   ```
   ==> dvdprime 3390 (server
   ```

   > **Important:** If you are installing from DVD, use the following command:
   >
   > ```
   > ==> dvdprime 3390 (dvd
   > ```

2. The command should complete quickly and you should see the following message:
```
HCPDVP8392I: DVDPRIME EXEC ENDED SUCCESSFULLY
```

Run INSTPLAN

1. Run the **`INSTPLAN DVD`** command to set up the configuration for the installation process. You should see the *z/VM INSTALLATION PLANNING* panel (see Figure 5-7):
```
==> instplan dvd
```

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

 Mark the product(s) selected to be installed into the filepool with an "F"
 and those selected to be installed to minidisks with an "M"
     M         VM             M         DIRM           M         ICKDSF
     M         OSA            M         PERFTK         M         RACF
     M         RSCS           M         TCPIP          M         VMHCD

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

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

 Enter the name of common service filepool.
     Filepool Name:       pool1__

 Select a System Type: Non-SSI or SSI (SSI requires the SSI feature)
     _ Non-SSI Install:     System Name ________
     x SSI Install:         Number of Members 2     SSI Cluster Name zvm63ssi
```
Figure 5-7 Installation planning panel

2. You might need to clear the window with the **Esc** key. You should then see the display as shown in Figure 5-7. It is recommended that you leave the “M”s in the top section alone.
3. 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. A file pool name of ***`pool1`*** is used in this example. Type the letter **x** next to `SSI Install`, set the number of members (**2** in this example), and choose a name for the cluster (***`zvm63ssi`*** in this example).
4. Press **F5** to proceed. You should see the SSI Cluster Installation panel. Press **F5** to accept the licensing terms.
5. You see the *z/VM INSTALLATION PLANNING PANEL 2*, as shown in Figure 5-8 on page 66. Answer *no* to the question about having your system managed by typing **n**. Press **F5** to continue.

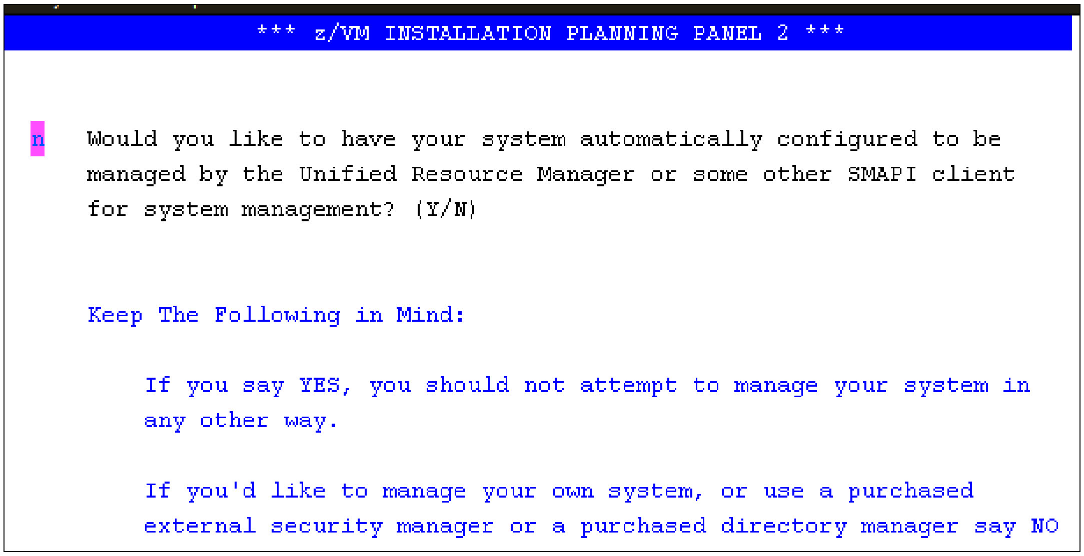


Figure 5-8 z/VM Installation Planning Panel 2

6. You should see the *z/VM INSTALLATION PLANNING PANEL 3* as shown in Figure 5-9. Enter the SSI member names and their corresponding logical partition (LPAR) names as seen on the HMC. Press **F5** to continue.

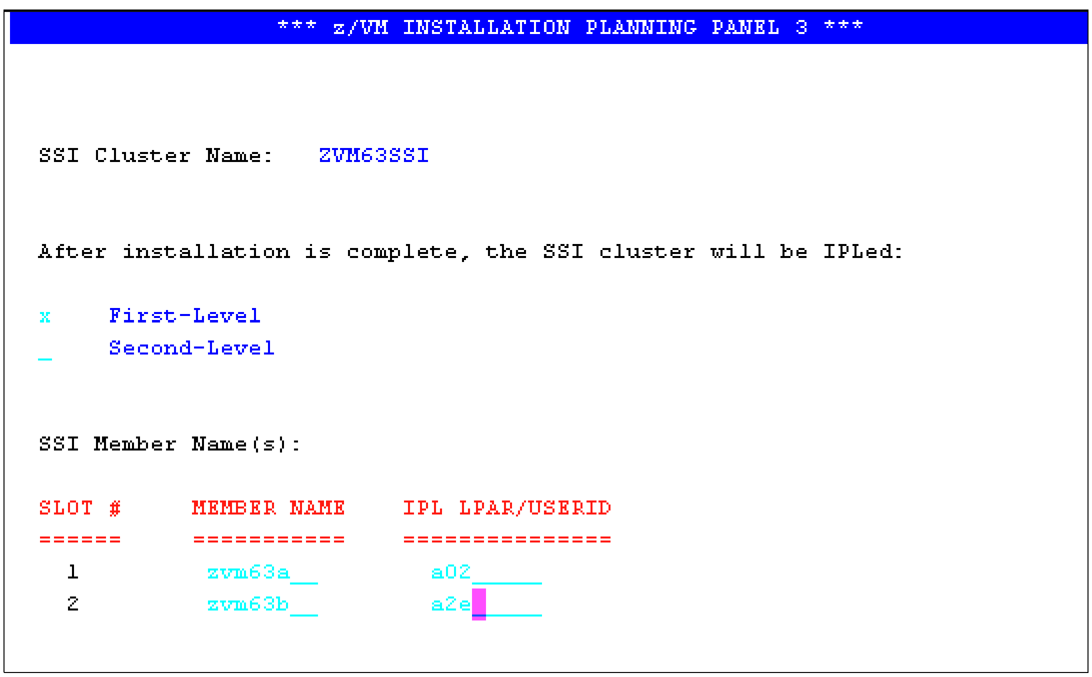


Figure 5-9 z/VM Installation Planning Panel 3

7. You will be shown a summary of your choices. If the values are correct, type **Y** to the question `DO YOU WANT TO CONTINUE?`
8. You should now see the *z/VM INSTALLATION VOLUME DEFINITION* panel as shown in Figure 5-10 on page 67.

```
               *** z/VM INSTALLATION VOLUME DEFINITION ***
       TYPE      LABEL      ADDRESS                     FORMAT (Y/N)
       COMMON    JV1036      1036                            _
       COMMON2   JV1037      1037
       RELVOL    JV1136      1136
       RELVOL2   JV1137      1137

       TYPE      LABEL      ADDRESS       TYPE      LABEL      ADDRESS
ZVM63A                              ZVM63B
       RES       JV1030      1030         RES       JV1130      1130
       SPOOL     JV1031      1031         SPOOL     JV1131      1131
       PAGE      JV1032      1032         PAGE      JV1132      1132
       WORK      JV1033      1033         WORK      JV1133      1133
       WORK      JV1034      1034         WORK      JV1134      1134
       WORK      JV1035      1035         WORK      JV1135      1135
```

Figure 5-10 z/VM Installation Volume Definition panel

9. Type in the volume labels and addresses from your worksheet. In this example, a prefix character of **J** is used. Press **F5** to continue.

10. You should see the *z/VM Installation First-level Configuration* panel as shown in Figure 5-11. The common volume addresses will almost always be identical. Enter the CTC device addresses.

```
              *** z/VM INSTALLATION FIRST-LEVEL CONFIGURATION ***

Real addresses for the common volume on each member LPAR:

  VOLUME     DASD        ZVM63A       ZVM63B
   TYPE      LABEL       ADDRESS      ADDRESS
  =======  ========    =========    =========
   COMMON    JV1036       1036         1036

CTC device addresses:

  From: ZVM63A                        From: ZVM63B
        To: ZVM63A      N/A                 To: ZVM63A    4120 5120
        To: ZVM63B    47e0 57e0             To: ZVM63B       N/A
```

Figure 5-11 z/VM Installation First-Level Configuration panel

11. Press **F5**. You should see a summary of your values, then the message:

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

12. Attach all DASD that will be part of the SSI cluster to `MAINT` with the **`ATTACH`** command. In this example, it is as follows:

```
==> att 1030-1037 1130-1137 *
15:10:13 1030-1037 ATTACHED TO MAINT
15:10:13 1130-1137 ATTACHED TO MAINT
```

Important: The devices ***1030-1037 1130-1137*** are in bold italics to signify that you should replace the example value with the correct value for your site. This convention is used throughout the book.

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

```
==> install
HCPIIS8490I NOW FORMATTING VOLUME 1036 (1 OF 16)
...
```

14. Finally, you should see the message `HCPMLP8392I INSTALL EXEC ENDED SUCCESSFULLY`.

    It is imperative that the **INSTALL** EXEC succeeds. If it does not, you must go back and fix it.

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

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

16. Run the **SHUTDOWN** command. This shuts down the SSI member that was IPLed last. You should see the system going down ending in a disabled wait with a state code of 961:

```
==> shutdown
...
HCPGIR450W CP entered; disabled wait PSW 00020000 00000000 00000000 00000961
```

    You should see the system identifier in the lower right go back to `IBMVMRAM` - the in-memory copy of z/VM.

17. Shutdown the in-memory system:

```
==> shutdown system ibmvmram
16:03:37 SYSTEM SHUTDOWN STARTED
```

    The in-memory copy of z/VM will now be halted on SSI member 1. The LPAR icon should turn red on the HMC.

Congratulations. You should now have z/VM 6.3 installed.

5.3.3 IPL the first SSI member

IPL your initial z/VM SSI system from DASD. Your *3270 Integrated Console* session should still be running. To do so, perform the following steps:

1. On the HMC, the LPAR of the first SSI member should still be selected. Click the **Tasks** drop-down menu in the upper right, then the **Recovery** slide-right menu, then the **Load** menu item.
2. The *Load* window opens. Follow these steps:
   a. Set the *Load Address* to the new system residence volume, which is ***1030*** in this example.
   b. Set the *Load Parameter* to **SYSG**. This specifies to use the Integrated 3270 console.
   c. Click **OK**.
3. When you see the *Load Task Confirmation* window, click **Yes**.
4. After a minute or less you should see a status of *Success* in the *Load Progress* window. Click **OK**.
5. Move back to the Integrated 3270 Console window. You should see the *Stand Alone Program Loader* panel as shown in Figure 5-12 on page 69. Press the **F10** key to continue the IPL of your z/VM system. It might take a while for the system to start IPLing.

```
STAND ALONE PROGRAM LOADER: z/VM VERSION 6 RELEASE 3.0

DEVICE NUMBER:    01030      MINIDISK OFFSET:   39          EXTENT:  1

MODULE NAME:      CPLOAD     LOAD ORIGIN:       1000

--------------------------------IPL PARAMETERS--------------------------------
fn=SYSTEM ft=CONFIG pdnum=1 pdvol=1036


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




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



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

Figure 5-12 Stand Alone Program Loader

6. At the `Start (Warm|Force|COLD|CLEAN)` prompt, enter **cold drain**:

   `==>` **`cold drain`**

7. At the `Change TOD` clock prompt, enter **no**:

   `==>` **`no`**

8. To the message, `To continue COLD start and delete files, enter GO`, type **go**:

   `==>` **`go`**

9. The first SSI member should IPL cleanly after about a minute. Disconnect from the `OPERATOR` virtual machine by using the **`DISCONNECT`** command:

   `==>` **`disc`**

The first SSI member should now be running.

5.3.4 IPL remaining SSI members

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

IPL each of the additional members from the HMC with the following steps:

1. Clear the previous LPAR and select the next one.
2. Start an *Integrated 3270 Console* from the *Recovery* menu, as shown in Figure 5-13 on page 70.

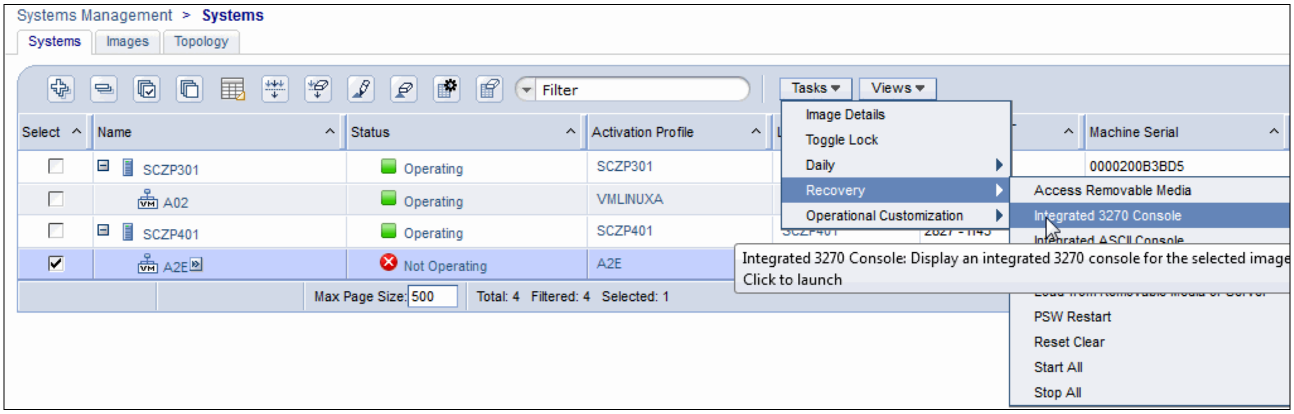


Figure 5-13 Starting a second Integrated 3270 Console

3. Invoke the **Load** operation from the same *Recovery* menu. You should see a window open, as shown in Figure 5-14.
4. In the *Load address* field, enter the real device address of the residence volume of the next LPAR (in this example it is **1130**). In the *Load Parameter* field, enter **SYSG**. Click **OK**.

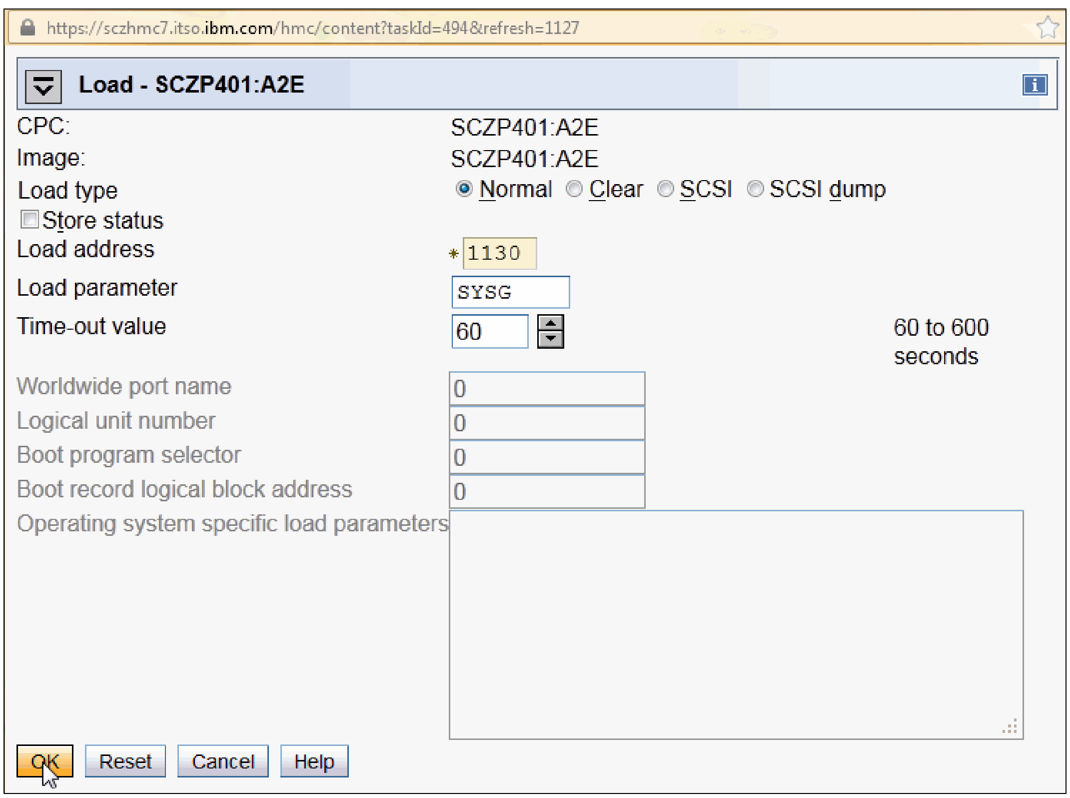


Figure 5-14 Loading z/VM on second LPAR

5. Switch to the Integrated 3270 Console window of the LPAR you are loading. After a short time, you should see z/VM coming up.

Important: You may see the following message:

```
HCPPLM1669I Waiting for ISFC connectivity in order to join the cluster.
```

This is *not good*. The member will likely wait forever to join. Check with the system administrator and verify that the CTCs are set up correctly and that you used the correct values. Verify that you typed the CTCs correctly. Figure 5-11 on page 67 shows a block diagram of the CTCs used in this example.

6. After a minute or two, when z/VM has finished IPLing, invoke the **DISCONNECT** command to disconnect from OPERATOR on each of the members:
```
==> disc
```

You should now have all of the members of the SSI running.

5.3.5 Verify the installation

Perform one more check to ensure that the RSU was installed:

- Log on to MAINT.
- Issue the QUERY CPLEVEL command to see the RSU level:

```
==> q cplevel
z/VM Version 6 Release 3.0, service level 1301 (64-bit)
Generated at 06/28/13 14:58:28 EDT
IPL at 09/04/13 09:12:18 EDT
```

Check your current RSU level and compare it to the latest available. See the following website:

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

If they do not match, you should apply the latest RSU as described in 6.1, "How to apply a recommended service upgrade" on page 105.

Congratulations. You should now have a z/VM 6.3 SSI cluster.

5.4 Configure TCP/IP

It is recommended that you initially configure TCP/IP using the **IPWIZARD** command on each of the SSI members. This wizard is generally used just once. After **IPWIZARD** creates the initial configuration files, they are typically maintained manually. A temporary Open Systems Adapter (OSA) triplet is used to initially get z/VM in the network. Later, the TCP/IP stack will be attached to the highly available VSWITCH.

To configure TCP/IP, perform the following steps:

1. From the HMC z/VM logon panel, **logon** to **MAINT**. The default password for MAINT is WD5JU8QP:

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

   You should see output similar to the following:.

```
LOGON MAINT
14:22:04 z/VM Version 6 Release 3.0, Service Level 0000 (64-bit),
```

```
14:22:04 built on IBM Virtualization Technology
14:22:04 There is no logmsg data
14:22:04 FILES: 0037 RDR,   NO PRT,   NO PUN
14:22:04 LOGON AT 14:22:04 EDT THURSDAY 06/06/13
14:22:04 GRAF L0005 LOGON  AS  MAINT    USERS = 16    FROM 9.57.139.102
z/VM V6.3.0    2013-06-04 12:50

DMSACP723I B (5E5) R/O
DMSACP723I D (51D) R/O
DMSACP723I E (551) R/O

*****************************************************************

THE MAINT630 USER ID **MUST** BE USED INSTEAD OF MAINT
WHEN INSTALLING SERVICE.

*****************************************************************

PRESS ENTER TO CONTINUE
```

2. When IPLing CMS before z/VM 6.2, the Enter key had to be pressed when the status area in the lower right reads "`VM READ`". Doing so allows the `PROFILE EXEC` to run. With z/VM 6.2 and later, the Enter key must be pressed a second time on certain virtual machines such as `MAINT`.

5.4.1 Use the IPWIZARD tool

The **`IPWIZARD`** tool enables you to quickly get z/VM onto a Internet Protocol network.

The **`IPWIZARD`** command is on the `MAINT 193` disk. You will need to access it on file mode `G` using the **`ACCESS`** command so that you will pick up **`IPWIZARD`** from that minidisk. To do this, perform the following steps:

1. Access the `MAINT 193` disk:

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

2. Invoke **`IPWIZARD`**.

```
==> ipwizard
```

3. The *z/VM TCP/IP Configuration Wizard* opens, as shown in Figure 5-15 on page 73. The first field, User ID, should always be **TCPIP**. Obtain the remaining values from section 2.9.4, "z/VM Networking resources" on page 32 and press **F8**.

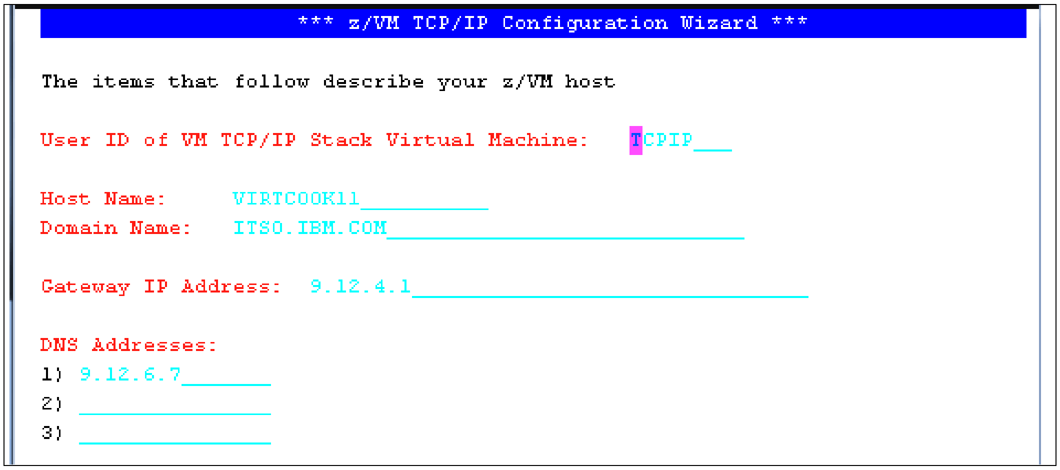


Figure 5-15 IPWIZARD panel 1

4. An *Interface Name* of **eth0** is arbitrary but recommended, as shown in Figure 5-16. The *Device Number* will be the starting address of the OSA triplet that the z/VM stack will use. The *IP Address* that must be routed to the OSA card will become the TCP/IP address of the z/VM system. The *Interface Type* will typically be **QDIO** with modern OSA devices. When completed, press **F8**.

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

Figure 5-16 shows the General Interface Configuration Panel.

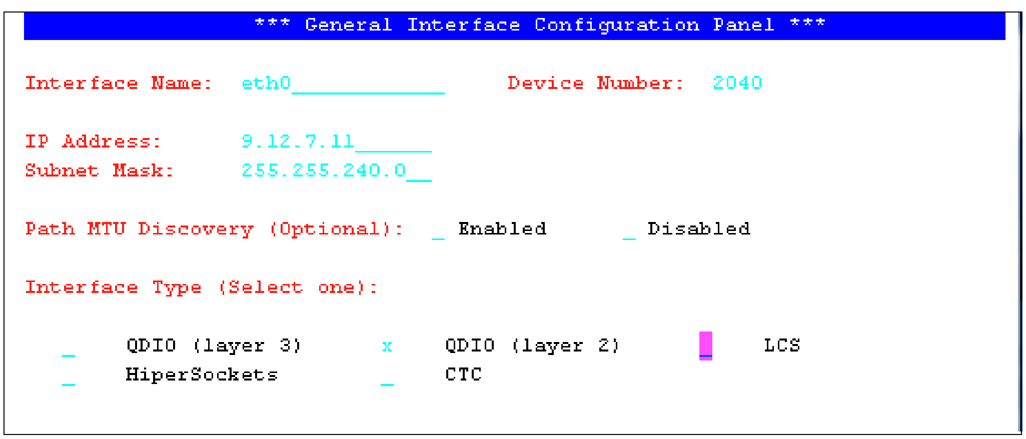


Figure 5-16 IPWIZARD panel 2

5. In general, a value for the *Port Number* is no longer necessary, as shown in Figure 5-17 on page 74. Press **F5** to complete the wizard.

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

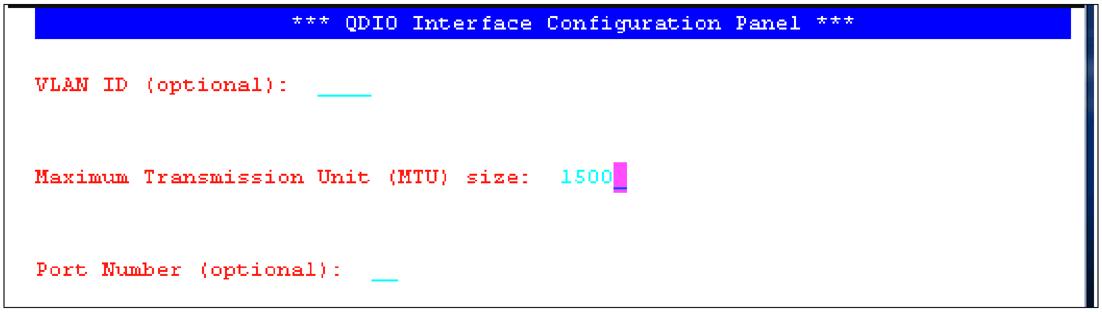


Figure 5-17 IPWIZARD panel 3

6. Enter **1** to restart the TCP/IP stack (you might see other warnings). Watch for the message `HCPINP8392I IPWIZARD EXEC ENDED SUCCESSFULLY`:

```
The TCP/IP stack (TCPIP) must be restarted as part of this procedure
Would you like to restart and continue?
Enter 0 (No), 1 (Yes) 1
USER DSC   LOGOFF AS  TCPIP    USERS = 10 FORCED BY MAINT
...
DTCIPW2519I Configuration complete; connectivity has been verified
DTCIPW2520I File PROFILE TCPIP created on TCPIP 198
DTCIPW2520I File TCPIP DATA created on TCPIP 592
DTCIPW2520I File SYSTEM DTCPARMS created on TCPIP 198
HCPINP8392I IPWIZARD EXEC ENDED SUCCESSFULLY
DMSVML2061I TCPIP 592 released
```

7. 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 Internet Protocol network and OSA information you were given are properly associated.
8. **Log off** from `MAINT` so that the `PMAINT 2CC` disk is freed up.
9. **Repeat the previous steps** for all other SSI members. When you run **IPWIZARD** on the other members, you should find that the network information is remembered.

All members of the SSI cluster should now be in the network.

> **HMC Integrated 3270 Console or 3270 emulator?** At this point, your SSI members should be accessible over the network. It is recommended to skip using the Integrated 3270 panel and access your new systems with a 3270 emulator. See 3.3, "3270 emulators" on page 42 for some brief words on that subject.
>
> To switch to a 3270 emulator, first **LOGOFF** of `MAINT` from the Integrated 3270 Console, but you could also **DISCONNECT**. If you log off, the session is ended. It is analogous to shutting and powering down a PC. If you disconnect, your session remains where it is and is resumed when you log back on. It is analogous to turning off a PC monitor. In general, you should **LOGOFF** of system administration virtual machines such as `MAINT`. However, you should always **DISCONNECT** from z/VM service machines such as `TCPIP` and virtual machines running Linux. Logging out of them terminates the service or crashes Linux.

5.5 Configure the XEDIT PROFILE

The **XEDIT** command looks for the file `XEDIT PROFILE` configuration file when it is invoked. Not all CMS virtual machines have a copy of this file, so XEDIT sessions look and behave differently. The `MAINT 191` (A) disk has a `PROFILE XEDIT` so when you are editing files on `MAINT`, the values in this profile are usually in effect.

If you have never used XEDIT before, there is a cheat sheet in Appendix A, "References and cheat sheets" on page 507. The z/VM 6.3 PDF library is on the web at the following site:

http://www-03.ibm.com/systems/z/os/zos/bkserv/zvmpdf/#zvm62

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

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

To configure the XEDIT profile on the SSI cluster, perform the following steps:

1. Log on to `MAINT` on the first SSI member if you are not already.
2. One default setting that can be dangerous, especially if you use `F12` to retrieve commands, is that PF12 is set to the **FILE** subcommand. Sometimes you might 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
   ```

   Before:

   ```
   SET PF12 FILE
   ```

   After:

   ```
   SET PF12 ?
   ```

3. Save your changes with the **FILE** subcommand.
4. Make the modified file available to other virtual machines by copying it to the `MAINT 19E` disk with file mode suffix `2`:

   a. Release the current `19E` disk:

   ```
   ==> rel 19e
   ```

   a. Link to the `MAINT 19E` disk read/write:

   ```
   ==> link * 19e 19e mr
   DASD 019E LINKED R/W; R/O BY    10 USERS
   ```

   a. Access the disk as file mode `F`:

   ```
   ==> acc 19e f
   ```

   b. Copy it to the `MAINT 19E` disk (`F`) with file mode suffix 2 (because the `MAINT 19E` disk is commonly accessed with a file mode suffix of `2`, files will not be seen by other virtual machines unless they have this file mode suffix):

   ```
   ==> copy profile xedit a = = f2
   ```

   c. Save the CMS named saved segment with the following commands:

   ```
   ==> acc 193 g
   ==> sampnss cms
   HCPNSD440I The Named Saved System (NSS) CMS was successfully defined in fileid 0
   002.
   ==> ipl 190 parm savesys cms
   HCPNSS440I Named Saved System (NSS) CMS was successfully saved in fileid 0002.
   ```

5. **Repeat the previous steps** on all other members in the SSI cluster.

The same `XEDIT PROFILE` should now be accessible to all virtual machines in the SSI cluster.

5.6 Customize the SYSTEM CONFIG file

The first configuration file read when z/VM IPLs is the `SYSTEM CONFIG` file. There is only one `SYSTEM CONFIG` file per SSI cluster.

The following changes are recommended:

- Increase retrieve key capacity
- Allow virtual disks (VDISKs) to be created
- Turn off the Disconnect Timeout feature (this will prevent idle disconnected users from being forced off the system)
- Define virtual switches to be used for Linux networking

To make these changes, perform the following steps:

1. Access the `PMAINT CF0` disk as read/write. Use the **LINK** command with multi-read (**MR**) parameter:

   ```
   ==> link pmaint cf0 cf0 mr
   ```

2. Use the **ACCESS** command to access it as your F disk:

   ```
   ==> acc cf0 f
   ```

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

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

4. Edit the original file:

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

5. Next, look for the `Features` statement:

   ```
   ====> /features
   ```

   You can search for it again or you can use **F8** to page down. The following changes and additions are recommended (see Example 5-1).

   – Increase the number of commands that can be retrieved from 20, to **99** (**1**).
   – Set the **Disconnect_Timeout** to **off** (**2**)so disconnected users do not get forced off.
   – Allow unlimited virtual disks to be created by users by changing `Userlim` to **infinite** (**3**) and by adding the **Syslim infinite** clause (**4**).

Example 5-1 Edit the system config file

```
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 , 1             /* 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 ,2 /* Don't force disconnected users */
  Vdisk ,                          /* Allow VDISKS for Linux swaps   */
```

```
    Syslim  infinite  , 4
 Userlim  infinite 3
```

6. For each SSI member, set real device equivalency IDs (EQIDs) for the OSA addresses to be used, and set the MAC address prefix. Real device mapping provides a means of identifying a device by EQID. This mapping ensures that virtual machines relocated by LGR continue to use the same or equivalent devices following a relocation. As shown in Example 5-2, use the **BOTTOM** subcommand to go to the bottom of the file (1). Add about 14 lines to the file (2).

   The VMLAN MACPREFIX statement will set the first three bytes of the MAC address created for each virtual NIC. If you have multiple z/VM systems, increment this value to avoid having identical MAC addresses created. In this example, ***02000B*** (3)and ***02000C*** (4)are used.

   The VMLAN TRANSIENT 0 statement prevents dynamic definition of Guest LANs by class G users.

Important: Regarding the setting of the VMLAN MACPREFIX value, the *CP Planning and Administration* manual states the following:

"In an SSI cluster, system-defined locally administered MAC addresses are created using the prefix value that is specified on the MACPREFIX operand. The MACPREFIX value must be different for each member of the cluster. The default value is 02xxxx, where xxxx is the member's slot number on the SSI statement. If the MACPREFIX value is explicitly defined, the VMLAN statement must be qualified for the member to which it applies. Therefore, if a VMLAN statement with the MACPREFIX operand is retained from the non-SSI system or created in this step, it must be qualified for member VMSYS01."

Example 5-2 Add EQIDs to the system config file

```
   ====> bot 1
   ====> a 14 2
   /* Add EQID statements for OSA addresses and unique MAC IDs */
   ZVM63A: begin
     rdev 2040-204f eqid osaset1 type osa
     rdev 4300-430f eqid osaset1 type osa
     vmlan macprefix 02000b 3
     vmlan limit transient 0
   ZVM63A: end

   ZVM63B: begin
     rdev 2040-204f eqid osaset1 type osa
     rdev 4300-430f eqid osaset1 type osa
     vmlan macprefix 02000c (4)
     vmlan limit transient 0
ZVM63B: end
```

7. Define two layer 2 virtual switches with the **DEFINE VSWITCH** statements. Modify the two starting addresses of the OSA triplets to those that you should have specified at the bottom of the worksheet in section 2.9.4, "z/VM Networking resources" on page 32. Follow the commands found here to insert lines and then add the lines found in Example 5-3 on page 78:

```
   ====> bot
```

```
====> a 4
```

Example 5-3 Define VSWITCHes

```
/* Define layer 2 VSWITCHes VSW1 and VSW2 */
define vswitch vsw1 rdev 4203 4300 ethernet
define vswitch vsw2 ethernet
```

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

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

9. Test your changes with the **CPSYNTAX** command, which is on the MAINT 193 disk. It must be run once for each member of the SSI cluster using the **LPAR** option to the **CPSYNTAX** command. Follow the commands listed here to accomplish this:

```
==> acc 193 g
==> cpsyntax system config f (lpar bvm1
CONFIGURATION FILE PROCESSING COMPLETE -- NO ERRORS ENCOUNTERED.
==> cpsyntax system config   f (lpar bvm2
CONFIGURATION FILE PROCESSING COMPLETE -- NO ERRORS ENCOUNTERED.
```

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

10. Release and detach the PMAINT CF0 disk with the **RELEASE** command:

```
==> rel f (det
DASD 0CF0 DETACHED
```

The SYSTEM CONFIG file should now be initially configured.

5.7 Configure additional network resources

In this section, we discuss the following recommended changes to the system:

- "Turn on the z/VM FTP server" on page 78
- "Shut down and re-IPL the SSI cluster" on page 79
- "Test changes" on page 81

The main TCP/IP configuration file is the PROFILE TCPIP file and is on the TCPMAINT 198 disk, which is accessed as the D disk.

5.7.1 Turn on the z/VM FTP server

Turn on the FTP server by performing the following steps:

1. Log on to TCPMAINT.
2. Make a backup copy of the TCP/IP configuration file, PROFILE TCPIP D:

```
==> copy profile tcpip d = tcpiorig = (oldd
```

3. Edit the TCP/IP configuration file:

```
==> x profile tcpip d
```

4. Make the following changes:

   a. Add three lines near the top and enter an AUTOLOG statement, a middle line for "FTPSERVE 0" that logs on the FTP server when TCP/IP starts, and the ENDAUTOLOG statement.

b. 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 TCP/IP is started.

The important lines before the file is edited and after are shown here:

Before:

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

After:

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

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

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

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

You can choose to test a shutdown and re-IPL of the SSI cluster now, or you can choose to first attach the z/VM TCP/IP stack to the highly available VSWITCH, which is described in the next section. If you choose to shut down the SSI cluster now, see section 5.7.2, “Shut down and re-IPL the SSI cluster” on page 79.

5.7.2 Shut down and re-IPL the SSI cluster

You can watch the z/VM member shut down and re-IPL from the *Integrated 3270 Console*. If you issue this command from a 3270 emulator, you will lose your session and will not see most of the shut down process. To shut down and re-IPL the SSI cluster, perform the following steps:

1. Log off from `MAINT` on all 3270 emulator sessions.
2. Start an *Integrated 3270 Console* session for the LPAR of the first SSI cluster member.
3. Log on to `MAINT`.

4. Using the **AT** command, issue the **SHUTDOWN** command for member 2. In this example, the system name is ***ZVM63B***:

```
==> at zvm63b cmd shutdown
...
```

5. If you have more than two members, repeat the previous **AT ... SHUTDOWN** step for those members.
6. From the HMC, start an *Integrated 3270 Console* session for the first member.
7. From member 1 of the cluster, issue the **SHUTDOWN REIPL** command:

```
==> shutdown reipl
...
```

   All members of the SSI cluster should now be down, and member 1 should be coming back up.
8. When the z/VM comes back up, you see messages as the system IPLs, and finally the z/VM logon panel. Log on as MAINT.
9. By default, the TCP/IP service virtual machine is not logged on (that will be customized later). Start TCP/IP with the **XAUTOLOG** command:

```
==> xautolog tcpip
Command accepted
AUTO LOGON  ***       EREP     USERS = 12
HCPCLS6056I XAUTOLOG information for TCPIP: The IPL command is verified by the I
PL command processor.
```

10. Try starting a 3270 emulator session to member 1. You should see a logon panel. If not, you will have to debug the problem from the *Integrated 3270 Console* session. For example, you could FORCE TCPIP and log on to TCP/IP interactively and watch for error messages.
11. Verify that TCP/IP is attached with the **QUERY VSWITCH** with **DETAILS** command:

```
==> q vswitch vsw1 det
VSWITCH SYSTEM VSW1     Type: QDIO    Connected: 1    Maxconn: INFINITE
  PERSISTENT  RESTRICTED    ETHERNET                  Accounting: OFF
  USERBASED
  VLAN Unaware
  MAC address: 02-00-0E-00-00-01    MAC Protection: Unspecified
  State: Ready
  IPTimeout: 5         QueueStorage: 8
  Isolation Status: OFF
 Uplink Port:
  RDEV: 7100.P00 VDEV: 0600 Controller: DTCVSW2
        EQID: OSASET1
    Uplink Port Connection:
      RX Packets: 45        Discarded: 0          Errors: 0
      TX Packets: 82        Discarded: 0          Errors: 0
      RX Bytes: 3330                TX Bytes: 12478
      Device: 0600  Unit: 000   Role: DATA       Port: 2049
 Adapter Connections:
    Adapter Owner: TCPIP    NIC: 0600.P00 Name: UNASSIGNED  Type: QDIO
      RX Packets: 5044      Discarded: 0          Errors: 0
      TX Packets: 82        Discarded: 0          Errors: 0
      RX Bytes: 220405              TX Bytes: 12478
      Device: 0600  Unit: 000   Role: DATA       Port: 0003
      Options: Ethernet Broadcast
        Unicast MAC Addresses:
          02-00-0E-00-00-05 IP: 9.60.18.144
        Multicast MAC Addresses:
```

```
01-00-5E-00-00-01
```

You should now have member 1 back up with TCP/IP attached to the highly available VSWITCH and the FTP server running.

IPL the other SSI members

You must now IPL the other SSI members. To do this, perform the following steps:

1. Go to the HMC and start an *Integrated 3270 Console* for the second SSI member.
2. IPL the LPAR with the **Load** task.
3. Go to the *Integrated 3270 Console* and complete the IPL of z/VM:
   a. Press **F10** at the SAPL window.
   b. Type **warm** at the `Start` command.
   c. Type **no** at the request to reset the TOD clock.
   d. If you are prompted for <something>, type **go**.
   e. Type **disc** to disconnect from `OPERATOR`. You should see a z/VM logon panel.
4. Disconnect from `OPERATOR` on the *Integrated 3270 Console*.
5. Log on as `MAINT`.
6. Start `TCPIP` with the **`XAUTOLOG`** command:

```
==> xautolog tcpip
Command accepted
...
```

7. If you have more than two members, repeat the previous step for those members.
8. Verify that the other nodes in the cluster can be accessed through the highly available VSWITCH.

The SSI cluster should now be back up.

5.7.3 Test changes

To test the changes that you made, perform the following steps:

1. Start a 3270 emulator session to the first SSI member.
2. Log on as `MAINT`.
3. Issue the **`QUERY SSI`** command:

```
==> q ssi
SSI Name: ZVM63SSI
SSI Mode: Stable
Cross-System Timeouts: Enabled
SSI Persistent Data Record (PDR) device: JV1036 on 1036
SLOT SYSTEMID STATE     PDR HEARTBEAT       RECEIVED HEARTBEAT
   1 ZVM63A   Joined    06/04/13   16:17:23 06/04/13   16:17:23
   2 ZVM63B   Joined    06/04/13   16:17:23 06/04/13   16:17:23
   3 -------- Available
   4 -------- Available
```

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

```
==> q retrieve
99 buffers available.  Maximum of 255 buffers may be selected.
==> q vdisk userlim
VDISK USER   LIMIT IS INFINITE
```

```
==> q vdisk syslim
VDISK SYSTEM LIMIT IS INFINITE,             0 BLK IN USE
```

5. Try starting an FTP session to all of the SSI members. You should get a logon prompt.

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

5.8 Add page and perm volumes

Each z/VM 6.3 SSI member is installed with one paging volume and one spool volume, either a 3390-3s or 3390-9s, depending on which type of disks the cluster was installed onto. One spool volume per member is probably adequate for Linux needs, however, more paging volumes are recommended.

Page and temporary disk volumes are not shared.

It is recommended that you add at least four 3390-3 paging volumes so you will have a total of five. Having adequate paging space will give you room 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 virtual machines combined. A second rule of thumb is to never allow your z/VM system's paging space to go above 50% utilized.

5.8.1 Format volumes for page space

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

Therefore, a REXX EXEC named **CPFORMAT** has been provided in the tar file associated with this book. It allows you to format many volumes with a single command. This EXEC can be found in Appendix A, "References and cheat sheets" on page 507"The CPFORMAT EXEC" on page 514. It is a wrapper around **CPFMTXA**. To use this EXEC, each DASD to be formatted must first be attached with the virtual device address and the same real device address (using **ATTACH *realDev* ***).

Note: This EXEC will label the volumes according to the convention described in section 2.3.1, "Volume labeling convention" on page 15. 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.

5.8.2 Copy the CPFORMAT EXEC to the members

Perform the following steps:

1. Log off from MAINT on the current member so you will be able to get the MAINT 191 disk in read/write mode using FTP.
2. Start an SSH session as root to the PC NFS server and **cd** to the /var/nfs/SG248147/vm/maint/ directory, which was created when you extracted the files associated with this book:

```
# cd /var/nfs/SG248147/vm/maint
```

3. List the files for the MAINT 191 disk:

```
# ls
callsm1.exec  cpformat.exec  ssicmd.exec
```

4. Start an FTP session to the first SSI member as MAINT. If you get a reply from the FTP server, it shows that it is configured correctly. Issue the **MPUT** subcommand to copy all files:

```
# ftp 9.12.7.12
Name (9.12.5.22:root): maint
331-Password: WD5JU8QP
230-MAINT logged in; working directory = MAINT 191
...
ftp> mput *
mput callsm1.exec [anpqy?]? a
Prompting off for duration of mput.
...
ftp> quit
```

   You should now have the **CPFORMAT EXEC**, (and two other files to be used later) on the MAINT 191 disk.

5. **Repeat the previous steps** for all other members in the cluster.

Each SSI member should now have access to the **CPFORMAT** EXEC.

5.8.3 Use the CPFORMAT EXEC

To use the **CPFORMAT** EXEC, perform the following steps:

1. Log in to **MAINT** on the first member.
2. List the files on the A disk with the **FILEccLIST** command:

```
==> filel
MAINT    FILELIST A0  V 169  Trunc=169 Size=10 Line=1 Col=1 Alt=0
Cmd   Filename Filetype Fm Format Lrecl    Records     Blocks   Date     Time
      CALLSM1  EXEC     A1 V         75        853          8  6/24/13 14:11:17
      CHPW620  XEDIT    A1 V        129        149          2  6/24/13 14:11:17
      CPFORMAT EXEC     A1 V         77        269          3  6/24/13 14:11:17
      SSICMD   EXEC     A1 V         64         71          1  6/24/13 14:11:17
      SSISHUTD EXEC     A1 V         72        104          2  6/24/13 14:11:17
      PROFILE  XEDIT    A1 V         45         17          1  6/24/13 11:12:06
      SETUP    $LINKS   A1 V         26         40          1  8/22/12  8:16:41
      PROFILE  EXEC     A1 V         72         32          1  8/07/12 10:49:50
      SYN      SYNONYM  A1 F         80          1          1  1/15/03  9:46:33
      PROFILE  XEDIORIG A1 V         45         17          1 11/18/98 12:26:20
```

3. Edit the file to set the first character that will be used in labels. Look for the variable **firstChar**.

```
==> x cpformat exec
====> /firstChar
/**************************************************************/
...
  Address COMMAND
  firstChar = 'J'
...
```

   If you would like the first character in the labels to be a letter other than J, change this setting.

4. 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 J<t><xxxx> where:
    <t> is type - P (page), M (perm), S (spool) or T (Temp disk)
    <xxxx> is the 4 digit address

Syntax is:
                <---------------<
   >>--CPFORMAT--.-vdev--------.--AS---.-PERM-.---------><
                 '-vdev1-vdev2-'       '-PAGE-'
                                       '-SPOL-'
                                       '-TEMP-'
```

The following example shows how to attach two 3390-3 volumes and use **CPFORMAT** to format them as paging space. Refer to the planning worksheets that should be completed in section 2.9.5, "z/VM DASD worksheet" on page 32.

5. The DASD that will be used for paging volumes on member 1 in this example are at real device addresses ***1038*** and ***103A***. Query the devices to see their status:

```
==> q 1038 103a
DASD 1038 NW1038  , DASD 103A NW103A
```

6. Attach the devices to MAINT by using the **ATTACH** command. This example uses the last parameter of *, which means the current virtual machine:

```
==> att 1038 103a *
DASD 1038 ATTACHED TO MAINT 1038 WITH DEVCTL HYPERPAV BASE
DASD 103A ATTACHED TO MAINT 103A WITH DEVCTL HYPERPAV BASE
```

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

```
==> cpformat 1038 103a as page
Format the following DASD:
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    1038 MAINT    1038 3390  NW1038 1038          0       3339
MAINT    103A MAINT    103A 3390  NW103A 103A          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    1038 MAINT    1038 3390  JP1038 1038          0       3339
MAINT    103A MAINT    103A 3390  JP103A 103A          0       3339
```

This formatting job should run for a number of minutes depending on many factors. Because these are being formatted as page, the **CPFORMAT EXEC** will also add owner information to the DASD. For this reason, *page volumes must be formatted on the SSI member on which they will be used*.

8. **Repeat the three previous steps on all other SSI members**. In this example, two more page volumes, ***1138*** and ***1139*** were added on the z/VM system ZVM63B.

5.8.4 Format DASD for minidisks

In addition to CP disks such as page space, system disks will be needed to create minidisks for the virtual machines. In this section the DASD which will be used for virtual machine minidisks will be formatted. Perform the following steps:

1. Start a 3270 session as MAINT on the first SSI cluster member.

2. Query the DASD that will be used for minidisks. In this example the DASD have real device addresses ***1262-126B*** and ***1362-1363***:

```
==> q 1262-126b 1362-1363
DASD 1262 NW1262  , DASD 1263 NW1263  , DASD 1264 NW1264  , DASD 1265 NW1265
DASD 1266 NW1266  , DASD 1267 NW1267  , DASD 1268 NW1268  , DASD 1269 NW1269
DASD 126A NW126A  , DASD 126B NW126B  , DASD 1362 NW1362  , DASD 1363 NW1363
```

3. Attach the volumes:

```
==> att 1262-126b 1362-1363 *
DASD 1262 ATTACHED TO MAINT 1262 WITH DEVCTL HYPERPAV BASE
DASD 1263 ATTACHED TO MAINT 1263 WITH DEVCTL HYPERPAV BASE
...
```

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

```
==> cpformat 1262-126b 1362-1363 as perm
Format the following DASD:
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    1262 MAINT    1262 3390  NW1262 1262          0      10017
MAINT    1263 MAINT    1263 3390  NW1263 1263          0      10017
...
WARNING - this will destroy data!
Are you sure you want to format the DASD as PAGE space (y/n)? y
...
DASD status after:
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    1262 MAINT    1262 3390  JM1262 1262          0      10017
MAINT    1263 MAINT    1263 3390  JM1263 1263          0      10017
MAINT    1264 MAINT    1264 3390  JM1264 1264          0      10017
MAINT    1265 MAINT    1265 3390  JM1265 1265          0      10017
MAINT    1266 MAINT    1266 3390  JM1266 1266          0      10017
MAINT    1267 MAINT    1267 3390  JM1267 1267          0      10017
MAINT    1268 MAINT    1268 3390  JM1268 1268          0      10017
MAINT    1269 MAINT    1269 3390  JM1269 1269          0      10017
MAINT    126A MAINT    126A 3390  JM126A 126A          0      10017
MAINT    126B MAINT    126B 3390  JM126B 126B          0      10017
MAINT    1362 MAINT    1362 3390  JM1362 1362          0      10017
MAINT    1363 MAINT    1363 3390  JM1363 1363          0      10017
```

You should now have many volumes that can be used for minidisks. Note that the labels are prefixed with **JM** in this example.

5.8.5 Update the SYSTEM CONFIG file

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

1. Log on to MAINT on member 1.
2. Link as read/write and access the PMAINT CF0 disk as file mode F:

```
==> link pmaint cf0 cf0 mr
==> acc cf0 f
```

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

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

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

```
==> x system config f
====> /page and
```

The following displays the pertinent information in the files before and after modification.

Before:

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

    ZVM63A:   BEGIN
                CP_Owned   Slot 255  JP1032
    ZVM63A:   END

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

    ZVM63B:   BEGIN
                CP_Owned   Slot 255  JP1132
    ZVM63B:   END
...
```

After:

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

    ZVM63A:   BEGIN
                CP_Owned   Slot 253  JP1038
                CP_Owned   Slot 254  JP103A
                CP_Owned   Slot 255  JP1032
    ZVM63A:   END

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

    ZVM63B:   BEGIN
                CP_Owned   Slot 253  JP1138
                CP_Owned   Slot 254  JP1139
                CP_Owned   Slot 255  JP1132
    ZVM63B:   END
...
```

5. Move down to the `User_Volume_List` section. User volumes (`PERM`) can be specified individually with the `User_Volume_List` statement, or with wildcards using the `User_Volume_Include` statement. If you are using the labeling convention enforced by the **`CPFORMAT EXEC`** ***and no other LPAR will be using the same volumes with the same prefix***, you should be able to use wildcards with the `User_Volume_Include` statement. In Example 5-4, all volume labels beginning with ***`JM1`*** will be attached to `SYSTEM` and be available for the creation of minidisks.

Example 5-4 Adding volumes to the system configuration file

```
====> /user_v
/*                            User_Volume_List                         */
/**********************************************************************/
/* These volumes contain the minidisks for your guests, as well as    */
/* the product disks for each installed release of z/VM in the SSI    */
/* cluster.  Volumes that hold "local" minidisks, i.e., minidisks     */
/* unique to a single member system, should be wrapped in BEGIN/END   */
/* statement.  If it becomes necessary to access a local minidisk     */
```

```
/* from a different member of the SSI cluster operating in REPAIR     */
/* mode, simply ATTACH the volume to SYSTEM.                          */
/**********************************************************************/

/**********************************************************************/
/* Shared User Volumes                                                */
/**********************************************************************/
                User_Volume_List  JV1037 JV1136 JV1137
                User_Volume_Include JM1*
...
====> file
```

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

```
User_Volume_List JM1262
User_Volume_List JM1263
User_Volume_List JM1264
User_Volume_List JM1265
User_Volume_List JM1266
...
```

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

```
==> acc 193 g
==> cpsyntax system config f (lpar bvm1
CONFIGURATION FILE PROCESSING COMPLETE -- NO ERRORS ENCOUNTERED.
==> cpsyntax system config f (lpar bvm2
CONFIGURATION FILE PROCESSING COMPLETE -- NO ERRORS ENCOUNTERED.
```

7. When you have confirmed that there are no syntax errors, you can release and detach the `PMAINT CF0`:

```
==> rel f (det
DASD 0CF0 DETACHED
```

You should now have volumes formatted for paging and minidisks.

5.8.6 Attach minidisk volumes to system

You could do a **SHUTDOWN** of the system to test your changes now or you can proceed. If you proceed, attach the volumes for minidisks to `SYSTEM` as shown here:

```
==> det 1262-126b 1362-1363
1262-126B DETACHED
==> att 1262-126b 1362-1363 system
DASD 1262 ATTACHED TO SYSTEM JM1262 HYPERPAV BASE
DASD 1263 ATTACHED TO SYSTEM JM1263 HYPERPAV BASE
DASD 1264 ATTACHED TO SYSTEM JM1264 HYPERPAV BASE
DASD 1265 ATTACHED TO SYSTEM JM1265 HYPERPAV BASE
DASD 1266 ATTACHED TO SYSTEM JM1266 HYPERPAV BASE
DASD 1267 ATTACHED TO SYSTEM JM1267 HYPERPAV BASE
DASD 1268 ATTACHED TO SYSTEM JM1268 HYPERPAV BASE
DASD 1269 ATTACHED TO SYSTEM JM1269 HYPERPAV BASE
DASD 126A ATTACHED TO SYSTEM JM126A HYPERPAV BASE
```

```
DASD 126B ATTACHED TO SYSTEM JM126B HYPERPAV BASE
DASD 1362 ATTACHED TO SYSTEM JM1362 HYPERPAV BASE
DASD 1363 ATTACHED TO SYSTEM JM1363 HYPERPAV BASE
```

The newly formatted volumes should now be attached to SYSTEM and available for use as minidisks.

5.9 Configure AUTOLOG1's PROFILE EXEC

When z/VM IPLs, normally the AUTOLOG1 virtual machine is logged on (unless the NOAUTOLOG parameter is specified at IPL). Its PROFILE EXEC is run when CMS IPLs. Using this file, perform the following tasks:

1. Configure Linux to shut down gracefully using the **SET SIGNAL** command.
2. Limit minidisk cache in main storage and turn it off in expanded storage with the **SET MDC** command.
3. Start virtual machines that should be started using the **XAUTOLOG** command.

Because AUTOLOG1 is now a multi-configuration virtual machine (IDENTITY), there is one virtual machine on each member. To configure the AUTOLOG1 PROFILE EXEC, perform the following steps:

4. Log on to AUTOLOG1.
5. Before pressing Enter at the VM READ prompt, type **acc (noprof** so that the **PROFILE EXEC** is not run:

```
LOGON AUTOLOG1
z/VM Version 6 Release 3.0, Service Level 1201 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES:   NO RDR,   NO PRT,   NO PUN
LOGON AT 13:32:04 EDT TUESDAY 09/03/13
z/VM V6.3.0    2013-06-24 16:58
==> acc (noprof
```

6. Make a copy of the original PROFILE EXEC:

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

7. Edit the PROFILE EXEC and add the following three lines below the Customer processing comment:

```
==> x profile exec
====> /customer
...
/* Customer processing can be added here                          */
/******************************************************************/
"CP XAUTOLOG TCPIP"             /* Autolog TCPIP                  */
"CP SET MDC STOR OM 256M"       /* Limit minidisk cache in CSTOR  */
"CP SET SIGNAL SHUTDOWN 600"    /* Allow guests 10 min to shut down */
...
====> file
```

8. **Perform the previous set of steps** on all other members in the SSI cluster.

The PROFILE EXEC on AUTOLOG1 191 disk should be configured for all members in the SSI.

5.9.1 Shut down and re-IPL the SSI cluster

It is recommended that you again shut down and re-IPL to test the changes. To do this, perform the following steps:

1. Log on to `MAINT` on the first SSI member.
2. Before you shut down, note that you have only one page volume (`JV6282` in Example 5-5) using the **`QUERY ALLOC PAGE`** command. A REXX EXEC has been provided to run any CP command on all members in the SSI cluster. It is named **`SSICMD EXEC`**. Use it to issue the **`QUERY ALLOC PAGE`** command across the SSI cluster:

   ```
   ==> ssicmd q alloc page
   ```

Example 5-5 Results of SSCMD EXEC

```
ZVM63A:
                EXTENT     EXTENT  TOTAL  PAGES   HIGH    %
VOLID  RDEV      START        END  PAGES IN USE   PAGE USED
------ ---- ---------- ---------- ------ ------ ------ ----
JV1032 1032          1       3338 600840     78     87   1%
                                  ------ ------        ----
SUMMARY                           600840     78          1%
USABLE                            600840     78          1%

ZVM63B:
                EXTENT     EXTENT  TOTAL  PAGES   HIGH    %
VOLID  RDEV      START        END  PAGES IN USE   PAGE USED
------ ---- ---------- ---------- ------ ------ ------ ----
JV1132 1132          1       3338 600840    128    135   1%
                                  ------ ------        ----
SUMMARY                           600840    128          1%
USABLE                            600840    128          1%
```

3. Shutdown and re-IPL the cluster again.

> **Important:** The **`SSICMD EXEC`** should not be used with asynchronous CP commands. The output might not be properly aligned with the member name prefix. Also, the **AT** command does not support commands that have an **`AT`** argument, such as **`QUERY`**.

4. In section 5.7.2, “Shut down and re-IPL the SSI cluster” on page 79, this task was accomplished manually:

   ```
   ==> shutdown
   ...
   ```

5. If you are using a 3270 emulator, you lose your session. If you watch the HMC, the SSI member LPARs should immediately turn from white to green, then return to white after a minute or so.
6. After the system comes back, log on as `MAINT`.
7. Use the **`SSICMD EXEC`** again to issue the **`QUERY ALLOC PAGE`** command across the SSI cluster:

   ```
   ==> ssicmd q alloc page
   ```

   You should now see that you have five paging volumes on each of the members, as shown in Example 5-6 on page 90.

Example 5-6 Results after running the second SSICMD EXEC

```
ZVM63A:
                 EXTENT     EXTENT  TOTAL  PAGES   HIGH    %
VOLID  RDEV      START        END  PAGES IN USE   PAGE USED
------ ---- ---------- ---------- ------ ------ ------ ----
JP1038 1038          0       3338 601020     72     72   1%
JP103A 103A          0       3338 601020      0      0   0%
JV1032 1032          1       3338 600840     53     63   1%
                                  ------ ------        ----
SUMMARY                            1761K    125          1%
USABLE                             1761K    125          1%

ZVM63B:
                 EXTENT     EXTENT  TOTAL  PAGES   HIGH    %
VOLID  RDEV      START        END  PAGES IN USE   PAGE USED
------ ---- ---------- ---------- ------ ------ ------ ----
JP1138 1138          0       3338 601020     64     75   1%
JP1139 1139          0       3338 601020     75     75   1%
JV1132 1132          1       3338 600840      1      1   1%
                                  ------ ------        ----
SUMMARY                            1761K    140          1%
USABLE                             1761K    140          1%
```

The output shows that there are three paging volumes on each SSI member constituting 1761 K pages, or about 6.9 GB of page space (a page is 4 KB). This is not much page space, but is sufficient for the relatively small setup that is described in this book. You will probably want to start with much more page space than this.

By now, you have created a new z/VM *user ID* or Single Configuration Virtual Machine (SCVM), `LNXMAINT`. Its main purpose is to provide a common CMS disk for all Linux virtual machines.

5.10 Create LNXMAINT for common files

In this section, we describe how to define your first z/VM virtual machine, `LNXMAINT`. It will be used to store files that will be shared by Linux virtual machines.

5.10.1 Define the user in the USER DIRECT file

A small 20-cylinder minidisk is allocated at virtual address `191` and a larger 500-cylinder minidisk (approximately 350 MB) to be shared by many guests, is defined at virtual address `192`. Use the next free DASD designated as PERM space on your worksheet (section 2.9.5, "z/VM DASD worksheet" on page 32). In this example, it is ***JM1262***. Cylinder 0 should always be reserved for the label; therefore, you should start minidisks at cylinder 1.

1. Make a copy of the original `USER DIRECT` file:

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

2. Edit the `USER DIRECT` file and add the following virtual machine definition to the bottom of the file. A comment is added signifying the split between z/VM system virtual machines and locally defined virtual machines (this can be helpful when moving to a new version of z/VM):

```
==> x user direct c
====> bottom
====> a 9
...
```

```
*------------------------------------------------------------
* z/VM system virtual machines are above, user defined below
*------------------------------------------------------------
USER LNXMAINT LNXMAINT 64M 128M G          1
 INCLUDE TCPCMSU                           2
 LINK TCPMAINT 592 592 RR                  3
 MDISK 0191 3390 0001 0020 JM1262 MR READ     WRITE    MULTIPLE    4
 MDISK 0192 3390 0021 0500 JM1262 MR ALL      WRITE    MULTIPLE    5
*                                                                  6
====> file
```

Note the following points for the numbers in black:

1 User ID `LNXMAINT` with password `LNXMAINT,default` size of 64 MB, with class 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 `JM1262`

5 Define `192` minidisk of size 500 cylinders (approximately 350 MB) with the special read password of **ALL**, which allows read access from any virtual machine without a disk password

6 An empty comment line for better readability

3. 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). z/VM allows overlaps to occur. That is, it allows the definition of multiple minidisks over the same disk space. This can cause problems. You can check for overlaps by using the **DISKMAP** command:

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

File USER DISKMAP A has been created.
```

4. 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
```

5. At the top of the file, you should see two overlap records in the `$$$$$$` volume. This is a dummy volume, so it is not a true overlap. Search for all other overlaps with the **ALL** subcommand:

```
====> all /overlap
         DATAMOV2    5F0     3380       00501       00501       00001 *OVERLAP*
--------------------  1  line(s) not displayed --------------------
         DATAMOV2    5FF     3380       00502       00502       00001 *OVERLAP*
--------------------  508  line(s) not displayed --------------------
```

You should see these two overlaps, which are expected.

6. Type **ALL** with no argument again to get out of this mode:

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

7. Now search for all the gaps using the **ALL** subcommand. You should also see some gaps:

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

```
-------------------- 5  line(s) not displayed --------------------
                                            0           0           1    GAP
-------------------- 198  line(s) not displayed --------------------
                                            0           0           1    GAP
-------------------- 298  line(s) not displayed --------------------

====> all
```

8. Three GAPs should be listed on the right side:
   - 501 cylinders on the $$$$$$ volume
   - 1 cylinder on the $$LNX1 volume
   - 1 cylinder on the $$LNX2 volume
   - 1 cylinder on volume used for LNXMAINT 191 and 192 disks (JM1262 in this example)

   You do not have to worry about the first three gaps because they are expected given the layout of the default USER DIRECT file. To avoid a 1-cylinder gap being reported on each user volume, it is recommended to use the virtual machine $ALLOC$. This user is set to NOLOG, which means it can never be logged on to. Thus it is not a conventional virtual machine, rather, it is a convenient place to put dummy minidisk definitions for cylinder 0 of all PERM volumes.
9. Exit the file USER DISKMAP with the **QUIT** command or by pressing **F3**.
10. Edit the USER DIRECT file again and add new minidisk definitions at virtual address A08-A15 for the first cylinder of each DASD volume that you added. There are 14 shown in our Example 5-7 after issuing the commands shown here:

```
==> x user direct
====> /user $alloc
```

Example 5-7 New minidisk definitions

```
MDISK A00 3390 000 001 JV1036 R
MDISK A01 3390 000 001 JV1037 R
MDISK A02 3390 000 001 JV1136 R
MDISK A03 3390 000 001 JV1137 R
MDISK A04 3390 000 001 JV1030 R
MDISK A05 3390 000 001 JV1033 R
MDISK A06 3390 000 001 JV1034 R
MDISK A07 3390 000 001 JV1035 R
MDISK A08 3390 000 001 JV1130 R
MDISK A09 3390 000 001 JV1133 R
MDISK A10 3390 000 001 JV1134 R
MDISK A11 3390 000 001 JV1135 R
MDISK A12 3390 000 001 JM1262 R
MDISK A13 3390 000 001 JM1263 R
MDISK A14 3390 000 001 JM1264 R
MDISK A15 3390 000 001 JM1265 R
MDISK A16 3390 000 001 JM1266 R
MDISK A17 3390 000 001 JM1267 R
MDISK A18 3390 000 001 JM1268 R
MDISK A19 3390 000 001 JM1269 R
MDISK A20 3390 000 001 JM126A R
MDISK A21 3390 000 001 JM126B R
```

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

```
==> diskmap user
```

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

File USER DISKMAP A has been created.
==> x user diskmap
====> prefix off
====> all /gap
                                            0         500         501    GAP
--------------------  8  line(s) not displayed --------------------
                                            0           0           1    GAP
--------------------  5  line(s) not displayed --------------------
                                            0           0           1    GAP
--------------------  497  line(s) not displayed --------------------
```

12. Quit XEDIT by pressing **F3**.

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

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

```
==> directxa user
z/VM USER DIRECTORY CREATION PROGRAM - VERSION 6 RELEASE 3.0
EOJ DIRECTORY UPDATED AND ON LINE
HCPDIR494I User directory occupies 105 disk pages
```

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

14. Log off from `MAINT` on member 1 to free up the `MAINT 2CC` disk.

15. Log on to `MAINT` on member 2. Run the **`DIRECTXA`** command on this member, then query the virtual machine again.

```
==> directxa user
z/VM USER DIRECTORY CREATION PROGRAM - VERSION 6 RELEASE 3.0
EOJ DIRECTORY UPDATED AND ON LINE
HCPDIR494I User directory occupies 105 disk pages
==> q lnxmaint
HCPCQU045E LNXMAINT not logged on
```

16. If you have more than one member, repeat the **`DIRECTXA`** step for all other members.

You have now defined your first z/VM virtual machine named `LNXMAINT` and brought it online to both SSI members.

5.10.2 Format the LNXMAINT minidisks

Now you should be able to log on to the new virtual machine and format its two minidisks. To do this, perform the following steps:

1. Log off from `MAINT`.
2. Log on to `LNXMAINT`.

```
LOGON LNXMAINT
z/VM Version 6 Release 3.0, Service Level 0000 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES:   NO RDR,   NO PRT,   NO PUN
LOGON AT 15:28:57 EDT TUESDAY 06/04/13
z/VM V6.3.0    2013-06-04 12:50

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 virtual machine, however, it has never been formatted as a CMS file system.

3. 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)
```

4. 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 500 cylinders formatted on D(192)
```

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

```
==> q disk
LABEL  VDEV M  STAT   CYL TYPE BLKSZ   FILES  BLKS USED-(%) BLKS LEFT  BLK TOTAL
LNX191 191  A   R/W    20 3390 4096        0          7-00       3593       3600
LXM192 192  D   R/W   500 3390 4096        0         13-00      89987      90000
MNT190 190  S   R/O   207 3390 4096      698      22295-60      14965      37260
MNT19E 19E  Y/S R/O   500 3390 4096     1126      29766-33      60234      90000
```

5.10.3 Create a PROFILE EXEC

Create a simple **PROFILE EXEC** that will be run each time this virtual machine is logged on.

1. Create the new file using XEDIT and add the following lines (be sure to type the **A** file mode so you do not pick up a `PROFILE EXEC` on another disk).

   REXX EXECs must always begin with a C language-style comment:

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

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

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

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

3. By default CMS tries to access the 191 disk as A and the 192 disk as D. Also you should have the TCPMAINT 592 disk accessed as E. Verify that these three disks are accessed with the **QUERY DISK** command:

```
==> q disk
LABEL  VDEV M  STAT   CYL TYPE BLKSZ   FILES  BLKS USED-(%) BLKS LEFT  BLK TOTAL
LXM191 191  A   R/W    20 3390 4096        2          9-01       3591       3600
LXM192 192  D   R/W   500 3390 4096        1         14-01      89986      90000
TCM592 592  E   R/O   140 3390 4096      890      16188-64       9012      25200
MNT190 190  S   R/O   207 3390 4096      698      22295-60      14965      37260
MNT19E 19E  Y/S R/O   500 3390 4096     1126      29766-33      60234      90000
```

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

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

5.10.4 Copy files associated with this book

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

1. **Log off of LNXMAINT** so that the 192 disk can be accessed as read/write.
2. **Start an SSH session** as root on the NFS server and change directory to the VM files associated with this book:

```
# cd /var/nfs/SG248147/vm/lnxmaint
```

3. List the files for the LNXMAINT 192 disk:

```
# ls
profile.exec  sample.conf-rh6  sample.parm-s11  swapgen.exec
rhel62.exec   sample.parm-rh6  sles11s2.exec
```

4. FTP to z/VM. By default, FTP copies files to your 191 disk, so first change the directory to the LNXMAINT 192 disk. The files are all in ASCII and the default behavior is to convert to ASCII to EBCDIC.

   Use the **mput *** subcommand to copy the files from the vm/ directory to LNXMAINT:

```
# ftp 9.12.7.11
User (9.12.7.11:(none)): lnxmaint
Password: WD5JU8QP
230-220-FTPSERVE IBM VM Level 630 at VIRTCOOK11.ITSO.IBM.COM, 15:34:46 EDT TUESDAY
2013-06-04
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
```

5. Log on to LNXMAINT.
6. Use the **FILELIST** command to show the files on the D disk:

```
==> filel * * d
LNXMAINT FILELIST A0  V 169  Trunc=169 Size=5 Line=1 Col=1 Alt=0
Cmd   Filename Filetype Fm Format Lrecl    Records     Blocks   Date     Time
      SLES11S3 EXEC     A1 V         63         12          1  9/03/13 13:07:21
      SWAPGEN  EXEC     A1 V         72        485          6  9/03/13 13:07:21
      SAMPLE   PARM-S11 A1 V         69         11          1  9/03/13 13:07:20
      SAMPLE   CONF-RH6 A1 V         38         13          1  9/03/13 13:07:19
      SAMPLE   PARM-RH6 A1 V         80          3          1  9/03/13 13:07:19
      RHEL64   EXEC     A1 V         63         14          1  9/03/13 13:07:18
      PROFILE  EXEC     A1 V         63         17          1  9/03/13 13:07:17
```

7. Logoff of LNXMAINT.

You should now have copied the files associated with this book to LNXMAINT 192.

5.11 Create identity LNXADMIN for Linux administration

Now it is time to create the first *Identity* or Multi-Configuration Virtual Machine (MCVM), LNXADMIN. An MCVM can be logged on to all members of the SSI at the same time. Therefore, it is not possible to migrate an MCVM between SSI members.

This virtual machine serves a number of administrative purposes:

- The Linux installation server: A file system *tree* of RPMs and other files that are required for installation are made available with NFS
- The clone server: for cloning from the golden image to target virtual machines (see Chapter 10, "Configure RHEL 6.4 for cloning" on page 193)
- The Red Hat kickstart server: for hosting the files necessary for automated installations (see Chapter 12, "Installing Red Hat Enterprise Linux with kickstart" on page 229)
- The administration server for other systems management tools such as xCAT.

In this section, you will learn how to perform an identity task, as shown in 5.11, "Create identity LNXADMIN for Linux administration" on page 96.

You will also define an SSI *identity*, also known as an *MCVM*. To do so, perform the following steps:

1. Log on to MAINT.
2. Determine the number of physical processors active with the **QUERY PROCESSORS** command:

   ```
   ==> q proc
   PROCESSOR 00 MASTER CP
   PROCESSOR 01 ALTERNATE CP
   ```

   In this example, there are two processors. This number will be used in the following steps.
3. Make a backup of the working USER DIRECT file:

   ```
   ==> copy user direct c = direwrks = (rep
   ```

4. Edit the USER DIRECT file:

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

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

5. 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, type the **""** prefix commands to duplicate the `IBMDFLT` profile on lines 38-50:

```
"" 38 *
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 *
"" 50 ******************************
```

6. Press **Enter** and the block will be duplicated.

7. Edit the duplicated profile by deleting the two `LINK MAINT 040x` lines, and inserting the lines that are shown in bold text:

```
PROFILE LNXDFLT
  COMMAND SET VSWITCH VSW1 GRANT &USERID
  COMMAND DEFINE NIC 600 TYPE QDIO
  COMMAND COUPLE 600 TO SYSTEM VSW1
  CPU 00 BASE
  CPU 01
  IPL CMS
  MACHINE ESA 8
  IUCV ALLOW
  OPTION CHPIDV ONE
  CONSOLE 0009 3215 T
  SPOOL 000C 2540 READER *
  SPOOL 000D 2540 PUNCH A
  SPOOL 000E 1403 A
  LINK MAINT 0190 0190 RR
  LINK MAINT 019D 019D RR
  LINK MAINT 019E 019E RR
  LINK LNXMAINT 0192 0191 RR
```

Notes for the preceding step 7:

- The three **COMMAND** lines give the virtual machine access to virtual switch VSW1 at logon time when the virtual machine is created. This precludes the need to add a VSWITCH GRANT statement each time that a Linux virtual machine is created.
- The two **CPU** lines define two virtual CPUs. It is recommended to set the number of virtual CPUs less than or equal the number of physical CPUs.
- The MACHINE statement sets the virtual machine type to ESA with a maximum of 8 CPUs. Even if your hardware does not have 8 IFLs, it is alright to set the maximum to 8 to leave *headroom*.
- The **IUCV ALLOW** line allows virtual machines to connect to other virtual machines, such as the Linux Terminal Server, by using IUCV.
- The **OPTION CHPIDV ONE** allows virtual machines to be relocated between SSI members.
- The last line provides read access to LNXMAINT 192 disk as the user's 191 disk.

8. Go to the bottom of the file and add the definition for a new identity named LNXADMIN. This virtual machine is given class B, in addition to the typical class G. This allows the virtual machine to run the **FLASHCOPY** command:

```
IDENTITY LNXADMIN LNX4VM 256M 1G BG
 INCLUDE LNXDFLT
 BUILD ON ZVM63A USING SUBCONFIG LNXADM-1
 BUILD ON ZVM63B USING SUBCONFIG LNXADM-2
 OPTION LNKNOPAS
SUBCONFIG LNXADM-1
 MDISK 0100 3390 0001 10016 JM1263 MR LNX4VM LNX4VM LNX4VM
 MDISK 0101 3390 0521  9496 JM1262 MR LNX4VM LNX4VM LNX4VM
SUBCONFIG LNXADM-2
MDISK 0100 3390 0001 10016 JM1362 MR LNX4VM LNX4VM LNX4VM
MDISK 0101 3390 0001 10016 JM1363 MR LNX4VM LNX4VM LNX4VM
```

The minidisks at virtual addresses 100 and 101 will be used to install Linux, for serving distribution install files, and other tasks.

9. Run the **DISKMAP** command, which creates a report file named USER DISKMAP.

```
==> diskmap user

==> x user diskmap
====> all /gap/|/overlap/
...
====> quit
```

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

```
==> directxa user
z/VM USER DIRECTORY CREATION PROGRAM - VERSION 6 RELEASE 3.0
EOJ DIRECTORY UPDATED AND ON LINE
HCPDIR494I User directory occupies 104 disk pages

EOJ DIRECTORY UPDATED AND ON LINE
```

You have now defined the virtual machine that will be the Linux administrative system.

5.11.1 Set LNXADMIN to start an IPL time

It is recommended that the new Linux administrative system be started at SSI IPL time. To do so, add an **XAUTOLOG** statement to the **PROFILE EXEC** on AUTOLOG:

1. Log on to MAINT, if you are not already.
2. Use the **LINK** and **ACCESS** commands to link and access the AUTOLOG1 191 disk read/write. This is the disk with the common EXEC that is run at IPL time for each member:

```
==> link autolog1 191 1191 mr
==> acc 1191 f
```

3. Edit the file **PROFILE EXEC**. Add a line to automatically start the LNXADMIN identity with the **XAUTOLOG** command:

```
==> x profile exec f   // add one line
...
"CP XAUTOLOG TCPIP"            /* Autolog TCPIP                 */
"CP SET MDC STOR 0M 256M"      /* Limit minidisk cache in CSTOR */
"CP SET MDC XSTORE 0M 0M"      /* Disable minidisk cache in XSTOR */
"CP SET SIGNAL SHUTDOWN 600"   /* Allow guests 10 min to shut down */
"CP XAUTOLOG LNXADMIN"         /* Start the Linux admin machine */
...
====> file
```

4. Release and detach the AUTOLOG1 191 disk with the **RELEASE** command:

```
==> rel f (det
DASD 1191 DETACHED
```

5. **Repeat the previous steps** for all other members in the SSI cluster.

The LNXADMIN identity should now be automatically started on all SSI cluster members.

5.12 z/VM security issues

This section briefly describes the following security issues:

- z/VM security products
- High-level z/VM security
- Linux virtual machine privilege classes
- z/VM virtual machine and minidisk passwords

VM security products

You might want to use a z/VM security product such as IBM RACF or CA VM:Secure. They allow you to address more security issues such as password aging and the auditing of users' access attempts. This book does not address the use of security products.

High-level z/VM security

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

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

Linux virtual machine privilege classes

Another security issue is the privilege class that Linux virtual machines are assigned. The IBM Redpaper™ publication *Running Linux Guests with less than CP Class G Privilege* addresses this issue.

The paper is available on the web at the following site:

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

5.12.1 Change passwords in USER DIRECT

In z/VM version 6.2 and earlier, all passwords were the same as the USER/IDENTITY ID, and minidisk passwords had different sets of values. This made it difficult to easily change all passwords. In z/VM 6.3, it is easier because the default `USER DIRECT` file has a single password for all USERs, IDENTITYs, and minidisks.

To modify all virtual machine and minidisk passwords to the same value, perform the following steps:

- Log on to `MAINT`.
- Make a backup copy of the `USER DIRECT` file:

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

- Verify 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`, do the following steps:

```
==> x user direct c
====> /lnx4vm
DMSXDC546E Target not found
====> quit
```

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

- Edit the `USER DIRECT` file use the change command:

```
==> x user direct c
====> c/WD5JU8QP/lnx4vm/* *
...
```

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

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

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

```
==> directxa user
z/VM USER DIRECTORY CREATION PROGRAM - VERSION 6 RELEASE 3.0
EOJ DIRECTORY UPDATED AND ON LINE
HCPDIR494I User directory occupies 103 disk pages
```

- **Repeat the `DIRECTXA` command** on all other members of the SSI.

Your new directory should now be online on both SSI members. Remember the new password.

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

5.13 Back up and restore your z/VM system

Your SSI system should now be customized with running TCP/IP stacks, two highly available virtual switches, a startup and shutdown process, and a virtual machine for shared CMS files. You should have changed the passwords. This would be a good time to back up the system to tape. To do so, refer to *Appendix E, Back up the z/VM system to Tape* in the manual *z/VM Installation Guide, version 6 release 2*, GC24-6246.

It is good to practice restoring a system. You do not want to be doing your first restore when the pressure is on. After you complete the backup, try to do a restore by following *Appendix H. Restore the z/VM system backup from tape* in the same manual.

If you do not have a tape device, there are also appendixes on backing up and restoring to and from DASD.

6

Service z/VM

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

— Albert Einstein

A new release of z/VM is made available approximately every 12 - 18 months. In addition to incorporating fixes to previously identified problems, new releases place an emphasis on new function and features that improve virtualization and the use of z/VM as a hypervisor for other System z Operating Systems. Clients are advised to run their production z/VM systems at the most current supported version/release available. IBM provides recommended maintenance service for all components, products, and features delivered with the z/VM base system in a single package called a *recommended service upgrade* (RSU). An RSU contains cumulative service in a prebuilt format that clients are advised to maintain RSU currency of a minimum of six months on their production z/VM systems.

RSUs ("stacked" or otherwise) are just packages, named `vrnn` - version, release, and a sequence number. For example, RSU `6204` is the fourth RSU for z/VM 6.2. You can get the latest RSU for a release by ordering special program temporary fix (PTF) number `UM97vr0`, where `vr` is the version and release. Inside the RSU is a collection of one or more service levels.

A *service level* (SL) is a pre-tested subset of all the available PTFs and is named `yynn`, where `yy` is the year of issue and `nn` is a sequence number. This sequence number has nothing to do with the RSU sequence number, so do not get upset if they do not match. Within each release, a single SL is established for the following parts of z/VM: The base (CP, CMS, and so on), TCP/IP, RACF, PerfKit, DIRMAINT, RSCS, HCD, and OSA/SF.

When it is time to deliver a new RSU, the RSU sequence number is incremented and all of the available service levels for that release are placed in it. At least one of them will be shiny new, but the others will be the same as on the previous RSU. Service levels are cumulative, containing in them all of the PTFs that were in the earlier service levels for that release.

This chapter describes how to apply the two main types of service:

- A *recommended service upgrade* (RSU), which is analogous to a *Service Pack.*
- A *program temporary fix* (PTF) which is analogous to a *bug fix*.

The process to install these types of service is basically the same.

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

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

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

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

- *z/VM Service Guide, version 6, release 1*, on the web at:

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

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

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

- 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 authorized program analysis report (APAR) has been applied (when used with STATUS).
- Creates PTF bitmap files (when used with BITMAP).

When **SERVICE** is successfully completed, the **PUT2PROD** EXEC places the z/VM components, features, or products that are installed on the z/VM System deliverable, and were serviced, into production. The following website is a good place to start:

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

The body of the page should look similar to the example that is shown in Figure 6-1 on page 105.

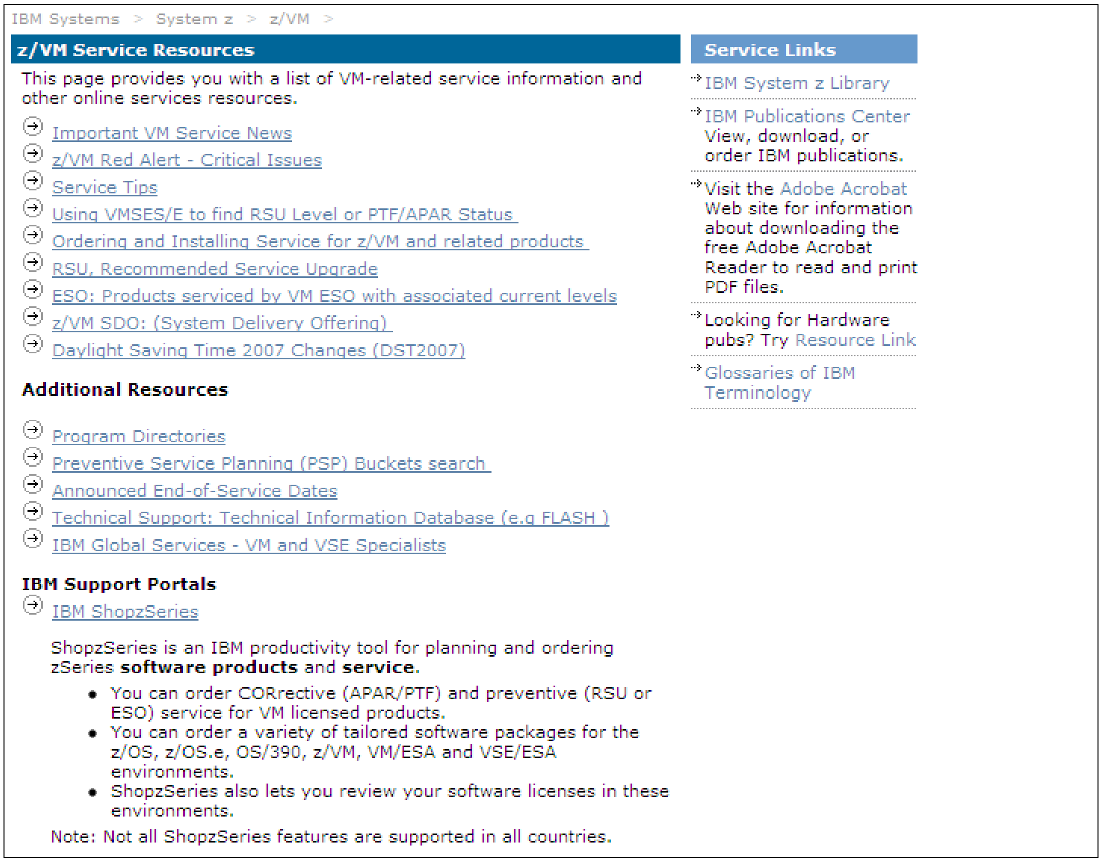


Figure 6-1 z/VM Service main Web page

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

The following sections comprise this chapter:

- "How to apply a recommended service upgrade" on page 105
- "How to apply a program temporary fix" on page 111
- "How to determine the service level of TCP/IP" on page 117

6.1 How to apply a recommended service upgrade

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).

The following website contains the latest RSU content information:

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

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

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

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

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

```
==> q cplevel
z/VM Version 6 Release 2.0, service level 1101 (64-bit)
Generated at 01/31/12 15:19:24 EDT
IPL at 06/15/12 10:30:33 EDT
```

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

Use the following overall steps in applying an RSU:

- “Get service from the Internet”
- “Download the service files” on page 107
- “Receive, apply, and build the service” on page 108
- “Put the service into production” on page 110

6.1.1 Get service from the Internet

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

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

Perform the following steps (note that these same steps are documented with some screen captures in 6.2, “How to apply a program temporary fix” on page 111):

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

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

- Click **IBM Shopz** under the *IBM Support Portals* section.
- Click the link **Sign In for registered users**, usually in the upper right. If you have a user ID and password, use that. If you do not, click the link **New user registration** and complete the form to create an ID and password. You must have your IBM customer number. (If you work for IBM, note the link *IBM employees must **sign in here***).
- Click the link **create new software orders** near the top.
- The *My Orders* page should show. Under the *Package Category* section, click the **z/VM - Service** radio button and also choose **RSU recommended service** in the drop-down menu. Click **Continue**.
- There will be five panels of forms that should be self-explanatory. On screen 3 of 5, choose the radio button that is applicable to your version of z/VM. In this example, **z/VM Version 6.2.0 Stacked 6202RSU (PTF UM97620)** was available.
- On screen 4 of 5, choose **Internet** as the delivery mechanism.
- On screen 5 of 5, complete the form and click **Submit**.
- In a few minutes, you should get two emails: One for the core RSU and one for the *PSP bucket* (additional fixes that might have come out after the RSU). Alternatively, you can click the refresh button on your browser. After some time, the *Status* should change to a link named **Download**, as shown in Figure 6-2 on page 107.

Figure 6-2 Downloading service directly from your browser

6.1.2 Download the service files

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

- Download the files to your desktop or another staging system. This example has two files: The SHIPTFSS file is for the PSP bucket, and the SHIPRSU1 file is for the RSU.
- FTP the file to the MAINT620 500 disk. Following is an example of FTPing from a DOS session:

```
C:\Downloads>ftp 9.12.7.12
User (9.60.18.249:(none)): maint620
Password:
ftp> cd maint620.500
250 Working directory is MAINT620 500
ftp> bin
200 Representation type is IMAGE.
ftp> quote site fix 1024
200 Site command was accepted.
ftp> put S9338801.shiptfss
...
ftp> put S9338766.shiprsu1
...
ftp> quit
```

- Log on to MAINT620.
- Access the MAINT620 500 disk as file mode C. Query the disks:

```
==> acc 500 c
DMSACC724I 500 replaces C (2CC)
==> q disk
LABEL  VDEV M  STAT   CYL TYPE BLKSZ   FILES  BLKS USED-(%) BLKS LEFT  BLK TOTAL
MNT191 191  A   R/W   175 3390 4096       26       231-01      31269      31500
MNT5E6 5E6  B   R/W     9 3390 4096      131      1265-78        355       1620
MNT500 500  C   R/W   900 3390 4096        2     50705-31     111295     162000
MNT51D 51D  D   R/W    26 3390 4096      299      1731-37       2949       4680
PMT551 551  E   R/W    40 3390 4096        9        92-01       7108       7200
MNT190 190  S   R/O   207 3390 4096      694     16694-45      20566      37260
MNT19E 19E  Y/S R/O   500 3390 4096     1126     29765-33      60235      90000
```

- List the files on the C disk and note the two new files:

```
==> listfile * * c
S1309082 SHIPRSU1 C1
6201RSU1 SERVLINK C1
S1309082 SHIPDOC  C1
```

- Deterse the documentation file, changing the file name prefix character to “d”:

```
==> deterse s1309082 shipdoc c d1309082 = =
```

- Deterse the RSU file changing the file type to SERVLINK (this step can take some time):

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

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

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

If this occurs, an extra step of creating a larger disk might be necessary.

6.1.3 Receive, apply, and build the service

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

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

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

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

- Log on to a 3270 session as MAINT620.
- Access the MAINT620 500 disk as C:

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

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

```
==> service all s1309082
...
VMFSUT2760I VMFSUFTB processing started
VMFSUT2760I VMFSUFTB processing completed successfully
VMFSRV2760I SERVICE processing completed with warnings
Ready(00004); T=*.**/*.** **:**:**
```

A return code of 0 is ideal. If the last Ready line has a number in parenthesis, that is the return code. In general, a return code of 4 is acceptable. That means that only warnings were issued. A return code of 8 or greater generally means that errors were encountered. View details with the **VMFVIEW SERVICE** command:

```
==> vmfview service
===> VMFVIEW - Message Log Browse of $VMFSRV  $MSGLOG  A1 <===
You are viewing ¬ST: messages from the LAST run.
Number of messages shown = 7 <===> Number of messages not shown = 764
*********************************************************************
****            SERVICE             USERID: MAINT620            ****
*********************************************************************
```

```
****            Date: 07/11/12           Time: 10:54:38            ****
************************************************************************
CK:VMFSUI2104I PTF UM33449 contains user information. Review the :UMEMO
CK:            section in file UM33449 $PTFPART
WN:VMFBDC2250W The following VMHCD objects have been built on BUILD0 300
WN:            (I) and should be copied to your workstation:
WN:VMFBDC2250W EEQINSTM MSIBIN
WN:VMFBDC2250W The following OSA objects have been built on BUILD0 100
WN:            (K) and should be copied to your workstation:
WN:VMFBDC2250W IOAJAVA  BIN
CK:VMFSRV1233I The following products have been serviced.
CK:VMFSRV1233I CMS CP OSA TCPIP VMHCD
```

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

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

```
==> ipl cms
DMSACC724I 19E replaces Y (19E)
DMSACP723I Y (19E) R/O
z/VM V6.2.0    2012-06-26 17:16
...
```

- Re-access the `MAINT 500` disk as `C`:

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

- Apply the PSP bucket if there is one. (In this example, there was no PSP bucket for RSU6202, so an older PSP bucket is shown):

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

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

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

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

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

- Press **F3** to get out of **`XEDIT`**.
- Log off from `MAINT620`.

6.1.4 Put the service into production

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

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

- Log on to MAINT620 on the first member.
- IPL CMS:

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

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

```
==> put2prod
...
VMFP2P1239I CP was serviced. Shutdown and re-IPL the system to employ the new
            service.
VMFP2P1239I CMS was serviced. Re-IPL CMS in all virtual machines running CMS to
            employ the new service.
VMFP2P2760I PUT2PROD processing completed successfully
```

- Review the messages with the **VMFVIEW PUT2PROD** command:

```
==> vmfview put2prod
You are viewing ¬ST: messages from the LAST run.
Number of messages shown = 4 <===> Number of messages not shown = 436
************************************************************************
****      PUT2PROD      SYSTEM: LEFT620       USERID: MAINT620      ****
************************************************************************
****           Date: 07/11/12           Time: 11:16:35              ****
************************************************************************
CK:VMFP2P1233I The following products have been put into production.
CK:            Recycle the appropriate servers.
CK:VMFP2P1233I CMS CP OSA TCPIP VMHCD
CK:VMFP2P1239I CP was serviced. Shutdown and re-IPL the system to employ
CK:            the new service.
CK:VMFP2P1239I CMS was serviced. Re-IPL CMS in all virtual machines
CK:            running CMS to employ the new service.
```

  In this example, the only messages are informational. If there are warning or error messages, those issues should be addressed.

- Press **F3** to get out of **XEDIT**.
- Even though the service has been "put into production", the **QUERY CPLEVEL** command should still return the current service level, in this example 1101 (the first RSU in the year 2011). This is because the new CP load module (nucleus) has not been loaded:

```
==> q cplevel
z/VM Version 6 Release 2.0, service level 1101 (64-bit)
Generated at 06/27/12 09:00:40 EDT
IPL at 06/27/12 09:34:06 EDT
```

- Perform the same PUT2PROD command on all other members of the SSI cluster.
- To load the new CP load module, you have to shut down and re-IPL the single system image (SSI) cluster:

– Log off from MAINT620.
– Log on to MAINT.
– Issue the **S REIPL** command:

```
==> shutdown
...
```

When your system comes back up, it should be at the new CP service level.

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

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

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

6.2 How to apply a program temporary fix

You might determine that you need to apply a specific fix or *program temporary fix* (PTF) to your system. For example, an APAR, VM65060, was opened when a problem was found with CMM.

The APAR was assigned the following PTF numbers for each of the following z/VM releases:

z/VM 5.4	UM33537
z/VM 6.1	UM33538
z/VM 6.2	**UM33539**

So for z/VM 6.2, you want to apply PTF **UM33539**. Following is an example of how to do so.

6.2.1 Get service using Shopz

Service for z/VM is still available on the media of tape. However, getting service over the Internet is more convenient and becoming more common. To do so, perform the following steps:

- Point a browser to the following URL:

  http://www14.software.ibm.com/webapp/set2/psearch/search?domain=sysz

- Enter the APAR number in the *Search For:* text field. In this example, the APAR is VM65060, and there was one hit, as shown in Figure 6-3 on page 112.

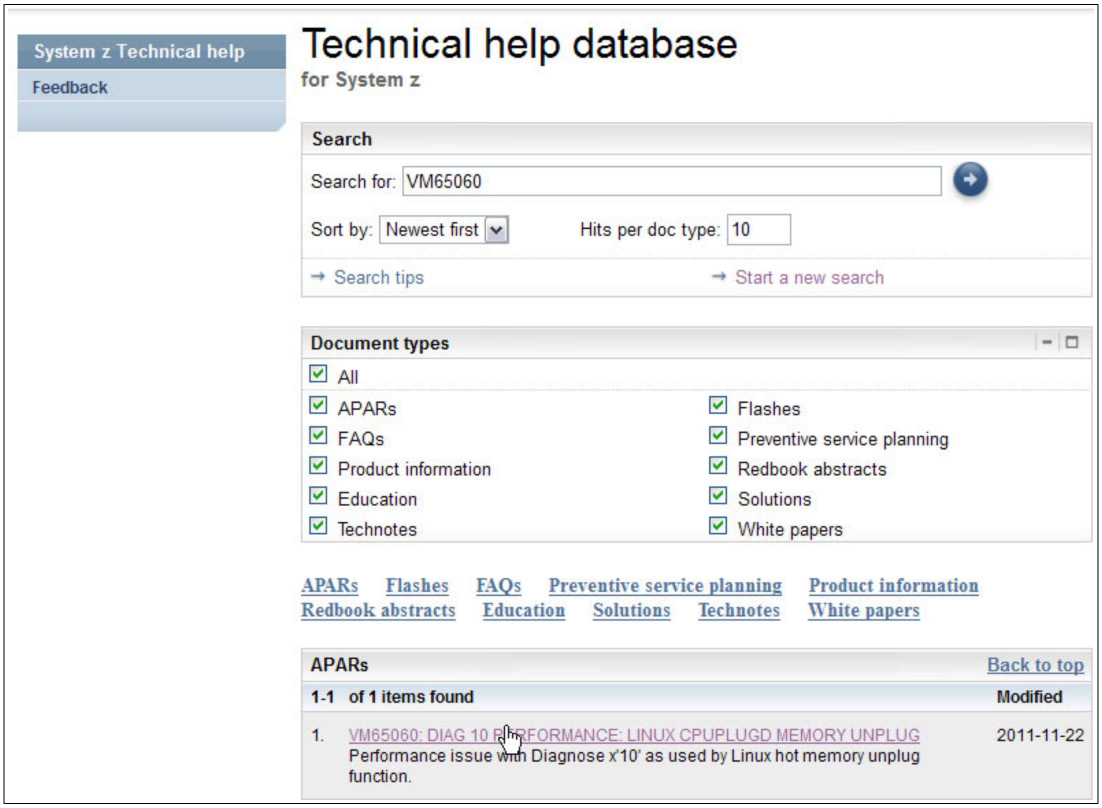


Figure 6-3 Searching for PTFs by APAR number

- Click the link of the APAR description.
- Farther down on the page, note the *Fixed component name*, which is important. In this example, it is **VM CP**.

At the bottom of the page the *Applicable component levels* section shows that PTF **UM33539** is available for z/VM 6.3. Before getting that PTF, you might want to be sure that it has not already been applied.

6.2.2 Determine if a PTF has been applied

Check to make sure that the PTF has not previously been applied. In this example, the PTF to check for is UM33539:

- Log on to `MAINT620`.
- Use the **`SERVICE ALL STATUS`** command followed by the PTF number to query whether it has been applied:

```
==> service all status um33539
VMFUTL2767I Reading VMFINS DEFAULTS B for additional options
VMFSRV2195I SERVICE ALL STATUS UM33539
VMFSRV2760I SERVICE processing started
DASD 0491 LINKED R/W; R/O BY    10 USERS
DASD 0492 LINKED R/W; R/O BY    10 USERS
DASD 019D LINKED R/W; R/O BY    17 USERS
DASD 0402 LINKED R/W; R/O BY    13 USERS
DASD 193C LINKED R/W; R/O BY    16 USERS
DASD 0200 LINKED R/W; R/O BY     2 USERS
DASD 0201 LINKED R/W; R/O BY PERSMAPI at ZVM63A
```

```
DASD 01CC LINKED R/W; R/O BY PERSMAPI at ZVM63A
DASD 029D LINKED R/W; R/O BY     2 USERS
VMFSRV1227I UM33539 is not received or applied
VMFSRV2760I SERVICE processing completed successfully
```

This shows that PTF UM33539 has *not* been applied. The sections that follow describe how to obtain and apply it.

6.2.3 Download the service to z/VM

From the previous APAR web page search, the link for **UM33539** is clicked, which results in a web page that should be similar to what is shown in Figure 6-4.

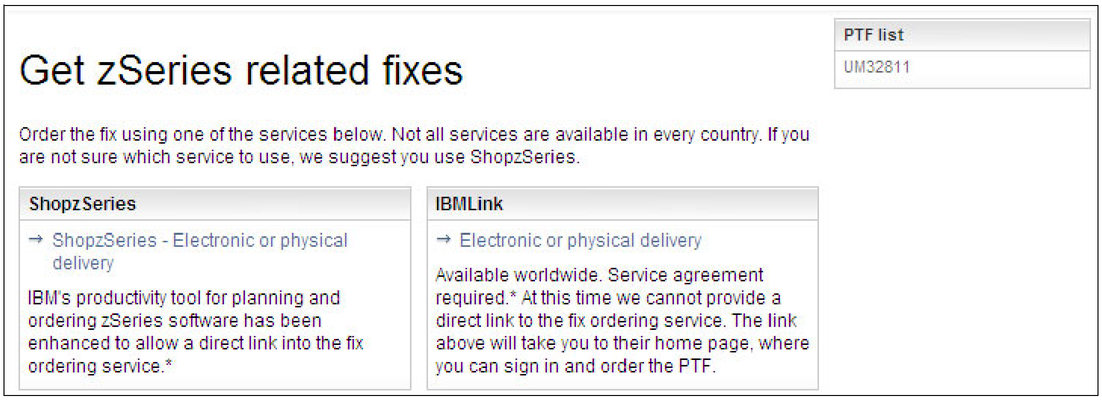


Figure 6-4   Getting fixes from Shopz on IBMLink

- In this example, the link **ShopzSeries - Electronic or physical delivery** is clicked.
- Sign in to Shopz with your IBM ID and follow the five self-explanatory steps to order your PTF (if you work for IBM, note the **sign in here** link). When you are finished, click **Submit** to place your order.
- You should receive an email within a few minutes. It will have your order number and a link to start the download of service files. Following is an example of the important information in the email:

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

- Point your browser to the link in the email. You should see a web page that looks similar to what is shown in Figure 6-5 on page 114.

Figure 6-5 Web page created for downloading a PTF

- Choose a method of downloading the **VMSES PTF Envelope** and the **VMSES Documentation Envelope** to a desktop or staging machine. In this example, **Download Director** was used.
- Copy both the SES and the documentation envelopes to z/VM in binary with fixed 1024-byte records to the `MAINT 500` disk. Usually, FTP is used. As you are downloading the files, note the file sizes. Following is an example of FTPing from a DOS session:

```
C:\downloads> ftp 9.12.7.12
User (9.60.18.249:(none)): maint620
Password:
...
ftp> cd maint620.500
250 Working directory is MAINT620 500
ftp> bin
200 Representation type is IMAGE.
ftp> quote site fix 1024
200 Site command was accepted.
ftp> mput s1041690.*
150 Storing file 'S1041690.SHIPDOCS'
250 Transfer completed successfully.
ftp: 6144 bytes sent in 0.00Seconds 6144000.00Kbytes/sec.
mput S1041690.SHIPTFSS? y
150 Storing file 'S1041690.SHIPTFSS'
250 Transfer completed successfully.
ftp: 10240 bytes sent in 0.00Seconds 10240000.00Kbytes/sec.
ftp> quit
```

- Log on to z/VM as `MAINT620`.
- Access the `MAINT620 500` disk as `C`:

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

- Verify that the files are there with the **LISTFILE** command:

```
==> listfile * * c
S1041690 SHIPDOCS C1
S1041690 SHIPTFSS C1
6201RSU1 SERVLINK C1
```

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

```
==> rename s1041690 shiptfss c = servlink =
==> rename s1041690 shipdocs c d1041690 servlink =
==> deterse s1041690 servlink c = = = (replace
==> deterse d1041690 servlink c = = = (replace
```

  Be sure that all commands complete successfully.

6.2.4 Receive, apply, and build service

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

To prepare to use the **SERVICE** command, you must have a minidisk with a lot of free space; that is what the MAINT620 500 minidisk is for.

- Access the MAINT620 500 disk as file mode C:

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

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

```
==> service all d1041690
...
VMFSUT2760I VMFSUFTB processing completed successfully
VMFSRV2760I SERVICE processing completed successfully
==> service all s1041690
...
VMFSRV1233I The following products have been serviced.
VMFSRV1233I CP
VMFSRV2760I SERVICE processing completed successfully
```

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

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

```
==> filel $vmf* $msglog
$VMFSRV  $MSGLOG  A1 V        80       1582         29  1/31/12 15:19:27
      $VMFBLD  $MSGLOG  A1 V        80        841         12  1/31/12 15:19:25
      $VMFAPP  $MSGLOG  A1 V        80        212          3  1/31/12 15:19:15
      $VMFREC  $MSGLOG  A1 V        80         69          1  1/31/12 15:19:15
      $VMFMRD  $MSGLOG  A1 V        80        270          4  1/31/12 15:19:14
      $VMFINS  $MSGLOG  A1 V        80        223          4 11/29/11  2:32:50
      $VMFP2P  $MSGLOG  A1 V        80       1741         32 11/29/11  0:55:22
```

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

```
==> vmfview service
===> VMFVIEW - Message Log Browse of $VMFSRV  $MSGLOG  A1 <===
You are viewing ¬ST: messages from the LAST run.
Number of messages shown = 2 <===> Number of messages not shown = 126
**********************************************************************
****            SERVICE              USERID: MAINT620             ****
**********************************************************************
****           Date: 01/31/12           Time: 15:19:13            ****
**********************************************************************
CK:VMFSRV1233I The following products have been serviced.
CK:VMFSRV1233I CP
```

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

6.2.5 Put the service into production

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

- Log on as MAINT620.
- IPL CMS:

```
==> ipl cms
z/VM V6.2.0    2011-11-15 11:26
```

- Access the VMSES/E test build disk as file mode B:

```
==> acc 5e6 b
DMSACC724I 5E6 replaces B (5E6)
```

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

```
==> put2prod
...
VMFP2P1239I CP was serviced. Shutdown and re-IPL the system to employ the new
            service.
VMFP2P2760I PUT2PROD processing completed successfully
```

  Note that the second to last message informs you that a **SHUTDOWN** and re-IPL is necessary. Again, watch for a return code of 0.

- Your PTF should now be *put into production*. You might or might not have to re-IPL the system, depending on the nature of the PTF applied. If you have to, be sure that you are in a position to re-IPL your system. You might want to shut down and re-IPL one member at a time with Live Guest Migrations of important Linux systems in between.
- Your z/VM system should come back in a few minutes. When the system comes back up, start a 3270 session to MAINT and again query the status of the PTF:

```
==> service cp status UM33539
VMFUTL2767I Reading VMFINS DEFAULTS B for additional options
VMFSRV2195I SERVICE CP STATUS UM33539
VMFSRV2760I SERVICE processing started
VMFSRV1226I CP (6VMCPR20%CP) PTF UM33539 status:
VMFSRV1226I    RECEIVED  01/31/12 15:19:15
VMFSRV1226I    APPLIED   01/31/12 15:19:15
VMFSRV1226I    BUILT     01/31/12 15:19:27
VMFSRV1226I    PUT2PROD  01/31/12 15:24:46 POKDEV62
VMFSRV2760I SERVICE processing completed successfully
```

- **Repeat the steps in this section** for all members in the SSI cluster.

This query shows that the PTF has been successfully applied.

6.2.6 Check for APARMEMO files

After you have applied PTFs, you should check for files with a file type of APARMEMO on the MAIN620T 500 disk. These files may have additional instructions on work to do after the PTFs have been applied. Perform the following steps:

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

```
==> acc 500 c
DMSACC724I 500 replaces C (2CC)
==> listfile * aparmemo c
6VMCPR20 APARMEMO C1
```

  In this example, there is one APARMEMO file.

- Look at the contents of the file:

```
==> type 6vmcpr20 aparmemo c

APAR MEMOS      01/30/12.14:16:55
=================================

        THE FOLLOWING MEMOS WERE INCLUDED WITH THE PTFS SHIPPED:

        NONE.
```

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

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

6.3 How to determine the service level of TCP/IP

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 following website:

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

Perform the following steps:

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

```
==> q vmlan
q vmlan
VMLAN maintenance level:
  Latest Service: Base
VMLAN MAC address assignment:
  System MAC Protection: OFF
  MACADDR Prefix: 02000B USER Prefix: 020000
  MACIDRANGE SYSTEM: 000001-FFFFFF
             USER:   000000-000000
VMLAN Unified Resource Manager status:
  Hypervisor Access: YES       Status: DISABLED BY SMAPI
  ID: NONE
  MAC Prefix: 02D737
```

```
VMLAN default accounting status:
  SYSTEM Accounting: OFF       USER Accounting: OFF
VMLAN general activity:
  PERSISTENT Limit: INFINITE   Current: 3
  TRANSIENT  Limit: INFINITE   Current: 0
```

The **Latest Service:** line shows that no APAR has been applied.

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

```
==> q vswitch vsw1
VSWITCH SYSTEM VSW1     Type: QDIO    Connected: 2    Maxconn: INFINITE
  PERSISTENT  RESTRICTED    NONROUTER                 Accounting: OFF
  USERBASED
  VLAN Unaware
  MAC address: 02-00-0B-00-00-01    MAC Protection: OFF
  State: Ready
  IPTimeout: 5        QueueStorage: 8
  Isolation Status: OFF
 Uplink Port:
  RDEV: 4203.P00 VDEV: 0600 Controller: DTCVSW1
        EQID: OSASET1
  RDEV: 4300.P00 VDEV: 0603 Controller: DTCVSW2  BACKUP
        EQID: OSASET1
```

This shows the controller is named `DTCVSW1`.

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

```
==> netstat tcp dtcvsw1 level
VM TCP/IP Netstat Level 620       TCP/IP Server Name: DTCVSW1

IBM 2818; z/VM Version 6 Release 2.0, service level 1101 (64-bit), VM TCP/IP Lev
el 620; RSU 0000 running TCPIP MODULE E2 dated 09/30/11 at 06:55
TCP/IP Module Load Address:  00C15000
```

This shows information about the current `TCPIP MODULE`.

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

```
==> tcpslvl tcpip module e
DTCLVL3306I SLVL data obtained; file TCPIP SLVLDATA A created
==> x TCPIP SLVLDATA
...
SLVL  TCPIP     ZVM620
...
SLVL SLVL  TCTOOSD   ZVM620
...
```

6.4 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.

7

Install a z/VM non-SSI LPAR

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

— Albert Einstein

> **Important:** These are not the official instructions on how to install z/VM 6.3. The program directory, installation manual, and other documents can be found at the following website:
>
> `http://www.vm.ibm.com/progdir`
>
> From that page, there is a link to the *z/VM Installation Guide, version 6 release 3* at the following website:
>
> `http://www.vm.ibm.com/progdir/hcsk2c10.pdf`
>
> The set of official z/VM manuals can be found at the following website:
>
> `http://publib.boulder.ibm.com/cgi-bin/bookmgr/Shelves/hcsh2ac0`
>
> This chapter describes installing z/VM 6.3 from an FTP server onto a single LPAR residing on DASD. It also addresses installing from DVDs. If you are installing onto SCSI disks or using significantly different parameters, you should use the official z/VM documentation.

This chapter consists of the following sections that should be completed in their entirety because the chapters that follow rely on these changes:

z/VM can be ordered and delivered electronically through IBM *Shopz* and made available via an FTP server. If you are taking this path, perform the steps in the following sections:

1. 5.1, "Obtain z/VM through electronic download" on page 58
2. 5.2, "Configure an FTP server for z/VM installation" on page 59

This section assumes that you have access to the z/VM 6.3 installation code in electronic format, however, there are comments describing small differences if you are using physical DVD media.

7.1 Install z/VM from DVD or FTP server

The sections that follow assume a *first-level* installation of z/VM from DVD or FTP server onto 3390 DASD. If you have not already done so, complete the worksheets in section 2.9, "Blank planning worksheets" on page 28.

If you are installing z/VM at the *second-level* (z/VM under z/VM) or onto FCP/SCSI disk, use this z/VM manual because the sections that follow do not address these options.

Starting a non-single system image (SSI) installation is identical to an SSI installation. Therefore, to start the installation, refer to section 5.3.1, "Start the z/VM installation" on page 61, and return to this chapter when you have an in-memory z/VM 6.3 system running.

7.1.1 Copy a vanilla z/VM system to DASD

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

- Move to the *Integrated 3270 Console* window. The RAMdisk should IPL and you should see z/VM boot as shown in Figure 7-1 on page 121. If the *Integrated 3270 Console* window is still blank, be patient; it might take a minute or two to initialize.

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

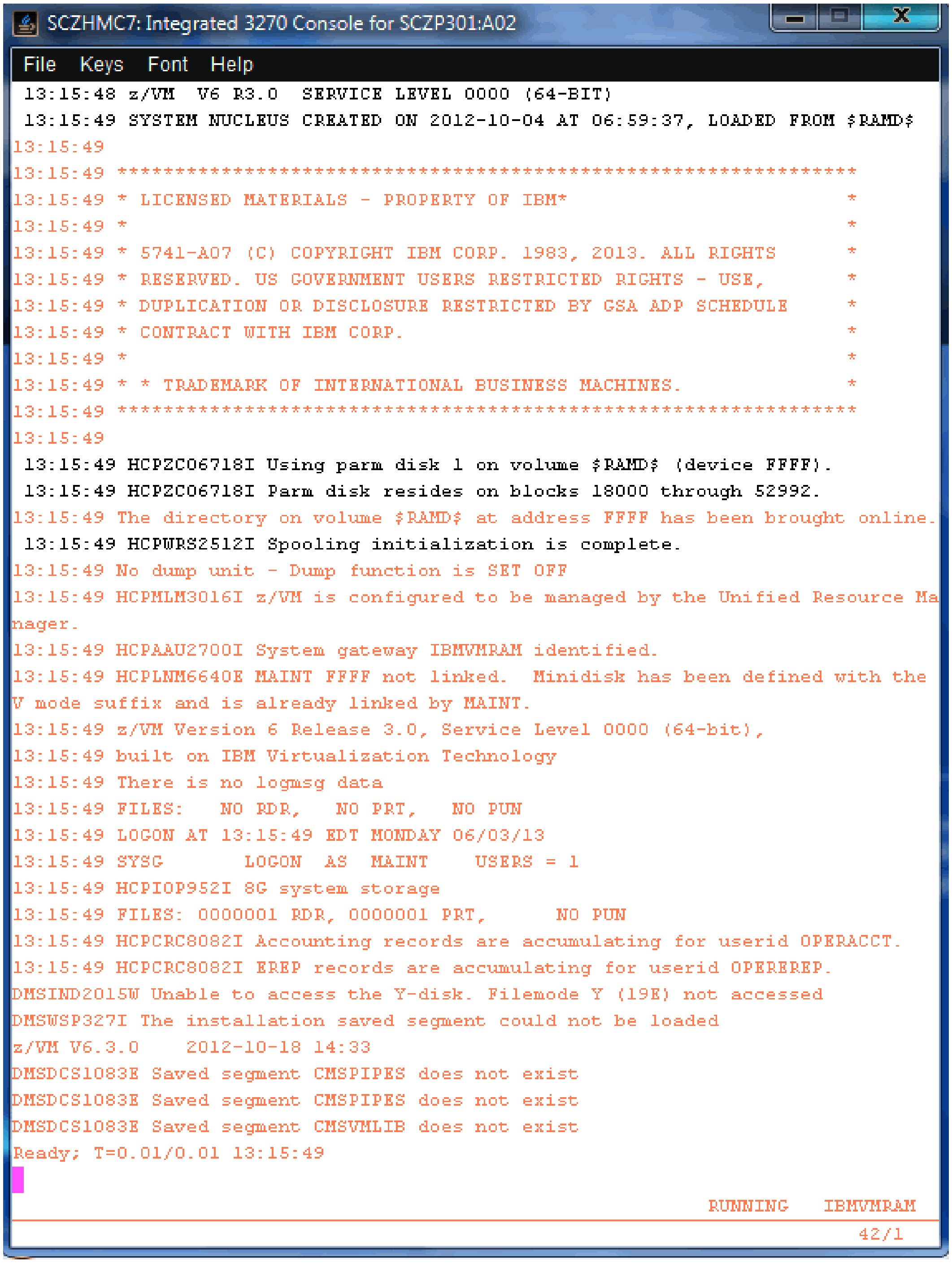


Figure 7-1 First z/VM 6.3 installation window

Run DVDPRIME

In this step, you run the **`DVDPRIME`** command. The format is **`dvdprime`** ***`dasdtype`*** **`(`*****`source`***.

In this example, the ***`dasdtype`*** is **`3390`** and the ***`source`*** is **`server`** - for FTP server.

```
==> dvdprime 3390 (server
```

The command should complete quickly and you should see the following message:

```
HCPDVP8392I: DVDPRIME EXEC ENDED SUCCESSFULLY
```

Run INSTPLAN

Next, run the **`INSTPLAN DVD`** command to set up the configuration for the installation process. You should see the *z/VM INSTALLATION PLANNING* panel:

```
==> instplan dvd
```

You might need to clear the screen with the **Esc** key. You should then see the display as shown in Figure 7-2. It is recommended that you leave the “M”s in the top section alone.

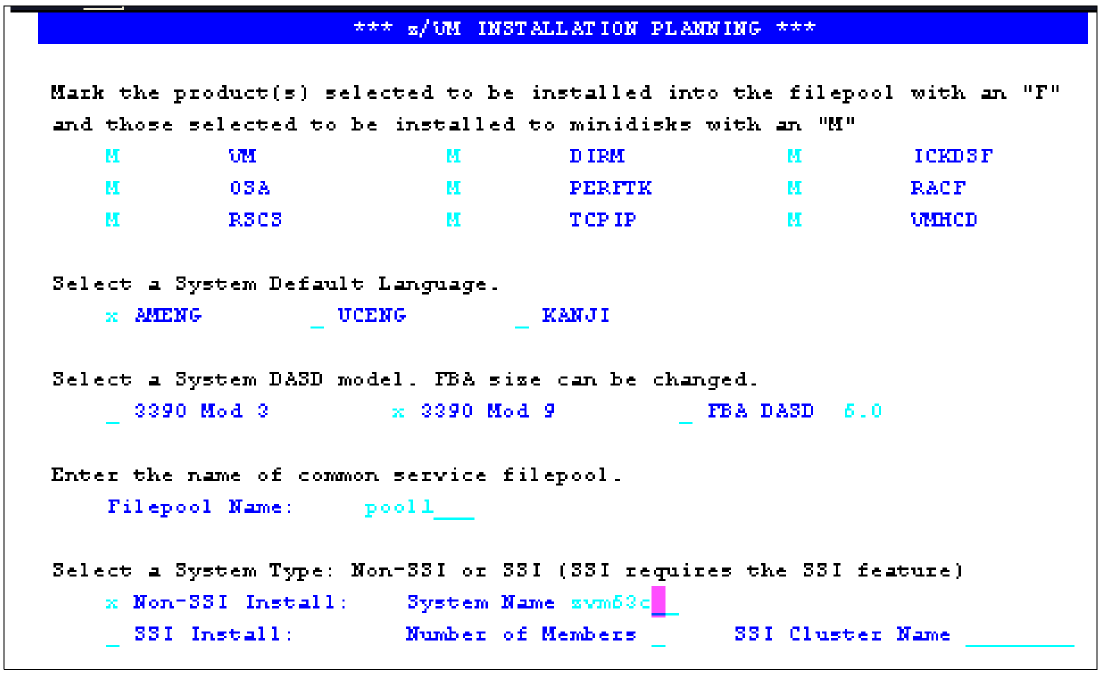


Figure 7-2 Installation planning panel

- Type the letter **x** next to `AMENG` (or select your language) and `3390 Mod 9`(or the type of DASD you will use) as shown above. A file pool name of ***`pool1`*** is used in this example. Type the letter **x** next to *Non-SSI Install*, and choose a name for the System (***`zvm63c`*** in this example).
- Press **F5** to proceed. You should see the *z/VM INSTALLATION PLANNING PANEL 2* as shown in Figure 7-3 on page 123. Answer **no** to the question about having your system managed by typing **n**. Press **F5** to continue.

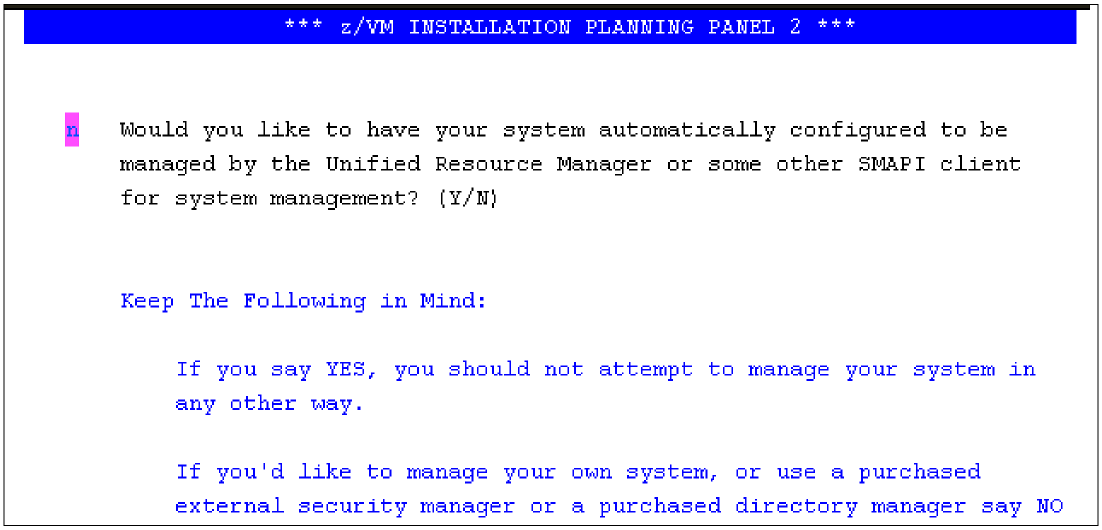


Figure 7-3 z.VM Installation Planning Panel 2

- A confirmation panel is presented that summarizes all options chosen. Select **Y** to continue.
- You should now see the *z/VM Installation Volume Definition* panel, as shown in Figure 7-4.

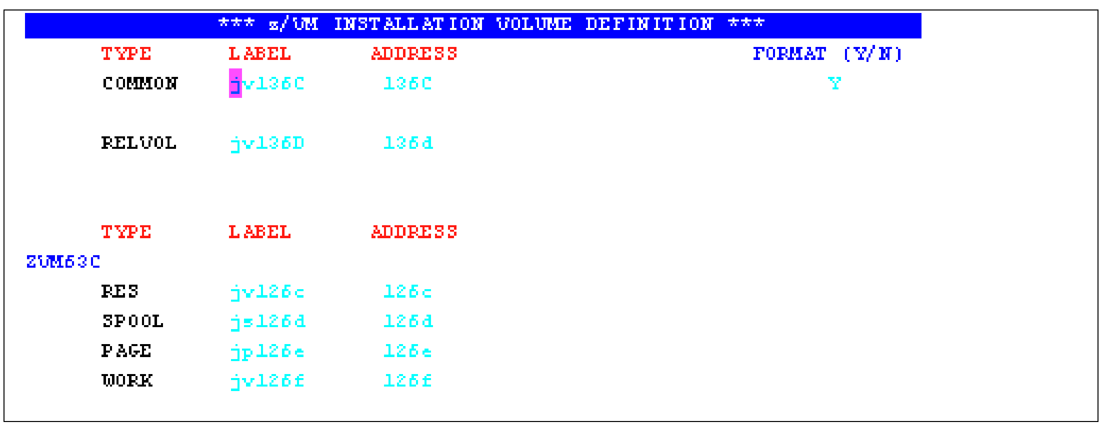


Figure 7-4 z/VM Installation Volume Definition panel

- Type in the volume labels and addresses from your worksheet. In this example, a prefix character of **K** is used.
- Press **F5**. You should see a summary of your values, then the message:

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

- Attach all DASD that will be part of the LPAR to `MAINT` with the **`ATTACH`** command. In this example, it is as follows:

```
==> att 126c-126f 136c-136d *
15:10:13 126C ATTACHED TO MAINT 126C WITH DEVCTL HYPERPAV BASE
...
```

Important: The devices ***`126c-126f 136c-136d`*** are in bold italics to signify that you should replace the example value with the correct value for your site. This convention is used throughout the book.

- Run the **INSTALL** command. The DASD will be formatted and the z/VM system disks will be copied. This step usually takes more than 1 hour:

```
==> install
HCPIIS8490I NOW FORMATTING VOLUME 136C (1 OF 6)
...
```

- Finally, you should see the messages:

```
hh:mm:ss HCPCRC8082I Accounting records are accumulating for userid DISKACNT
hh:mm:ss DISCONNECT AT hh:mm:ss timezone weekday mm/dd/yy
hh:mm:ss Press enter or clear key to continue
```

  It is imperative that the **INSTALL** EXEC succeeds. If it does not, you must go back and fix it.

- Press the **Enter** key.
- Log on as MAINT630 user (new common password for all users is **WD5JU8QP**).
- Run the **SHUTDOWN** command. This will shut down the freshly IPLed system. You should see the system going down ending in a disabled wait with a state code of 961:

```
==> shutdown
...
HCPGIR450W CP entered; disabled wait PSW 00020000 00000000 00000000 00000961
```

  You should see the system identifier in the lower right go back to IBMVMRAM - the in-memory copy of z/VM.

- Shut down the RAMDISK system:

```
==> shutdown system ibmvmram
16:03:37 SYSTEM SHUTDOWN STARTED
```

  The in-memory copy of z/VM will now be halted on the LPAR. The LPAR icon should turn red on the HMC.

7.1.2 IPL the z/VM LPAR

IPL your initial z/VM system from DASD. Your *3270 Integrated Console* session should still be running:

- On the HMC, the z/VM LPAR should still be selected. Click the **Tasks** drop-down menu in the upper right, then the **Recovery** slide-right menu, then the **Load** menu item.
- The Load window opens as shown in Figure 7-5 on page 125. Follow these steps:
  a. Set the *Load Address* to the new system residence volume, which is ***126C*** in this example.
  b. Set the *Load Parameter* to **SYSG**. This specifies to use the Integrated 3270 console.
  c. Click **OK**.
- When you see the *Load Task Confirmation* window, click **Yes**.
- After a minute or less you should see a status of *Success* in the *Load Progress* window. Click **OK**.
- Move back to the Integrated 3270 Console window. You should see the *Stand Alone Program Loader* panel, as shown in Figure 7-5 on page 125. Press the **F10** key to continue the IPL of your z/VM system. It might take a while for the system to start IPLing.

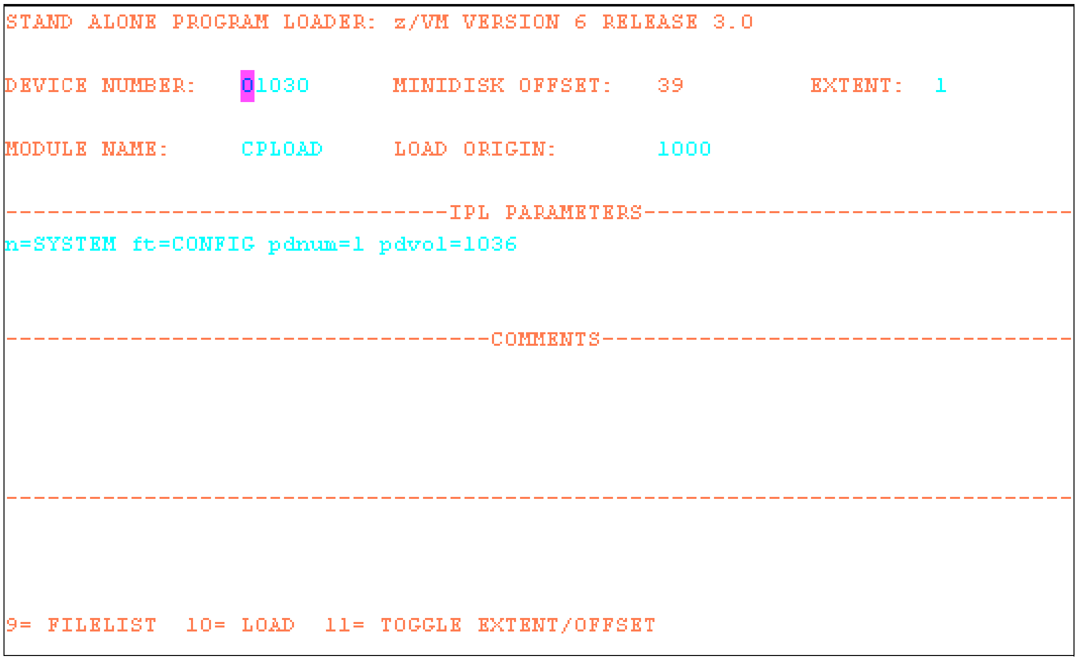


Figure 7-5 Stand Alone Program Loader panel

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

  `==>` **`cold drain`**

- At the `Change TOD` clock prompt, enter **no**:

  `==>` **`no`**

- To the message, `To continue COLD start and delete files, enter GO`, type **go**:

  `==>` **`go`**

- The system should IPL cleanly after about a minute. Disconnect from the `OPERATOR` virtual machine using the **`DISCONNECT`** command:

  `==>` **`disc`**

The non-SSI system should now be running.

7.1.3 Verify the installation

Perform one more check to ensure that the RSU was installed:

- Log on to `MAINT`.
- Issue the **`QUERY CPLEVEL`** command to see the RSU level:

```
==> q cplevel
z/VM Version 6 Release 3.0, service level 1301 (64-bit)
Generated at 06/28/13 14:58:28 EDT
IPL at 09/04/13 09:12:18 EDT
```

Check your current RSU level and compare it to the latest available at the following website:

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

If they do not match, apply the latest RSU as described in section 6.1, “How to apply a recommended service upgrade” on page 105.

Congratulations. You should now have a z/VM 6.3 LPAR.

7.2 Configure TCP/IP

It is recommended that you initially configure TCP/IP using the **IPWIZARD** command on the LPAR. This wizard is generally used just once. After **IPWIZARD** creates the initial configuration files, they are typically maintained manually. A temporary OSA triplet is used to initially get z/VM in the network. Later, the TCP/IP stack will be attached to the highly available VSWITCH.

To configure TCP/IP, perform the following steps:

- From the HMC z/VM logon panel, log on to `MAINT`. The default password for `MAINT` is `WD5JU8QP`:

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

  You should see output similar to that shown in Example 7-1.

Example 7-1 Logging on to MAINT and pressing Enter twice

```
LOGON MAINT
14:22:04 z/VM Version 6 Release 3.0, Service Level 1301 (64-bit),
14:22:04 built on IBM Virtualization Technology
14:22:04 There is no logmsg data
14:22:04 FILES:   NO RDR,   NO PRT,   NO PUN
14:22:04 LOGON AT 09:13:50 EDT WEDNESDAY 09/04/13
z/VM V6.3.0    2013-09-04 07:58

DMSACP723I B (5E5) R/O
DMSACP723I D (51D) R/O
DMSACP723I E (551) R/O

*****************************************************************

THE MAINT630 USER ID **MUST** BE USED INSTEAD OF MAINT
WHEN INSTALLING SERVICE.

*****************************************************************

PRESS ENTER TO CONTINUE
```

- When IPLing CMS before z/VM 6.2, the **Enter** key has to be pressed when the status area in the lower right reads “`VM READ`”. Doing so allows the `PROFILE EXEC` to run. With z/VM 6.2 and later, **Enter** must be pressed a second time on certain virtual machines such as `MAINT`.

7.2.1 Use the IPWIZARD tool

The **IPWIZARD** tool enables you to quickly get z/VM onto a Internet Protocol network.

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

- Access the `MAINT 193` disk:

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

- Invoke **IPWIZARD**:

  `==>` **`ipwizard`**

  You should see panels that are similar to what is shown in Figure 7-6, Figure 7-7, and Figure 7-8 on page 128.

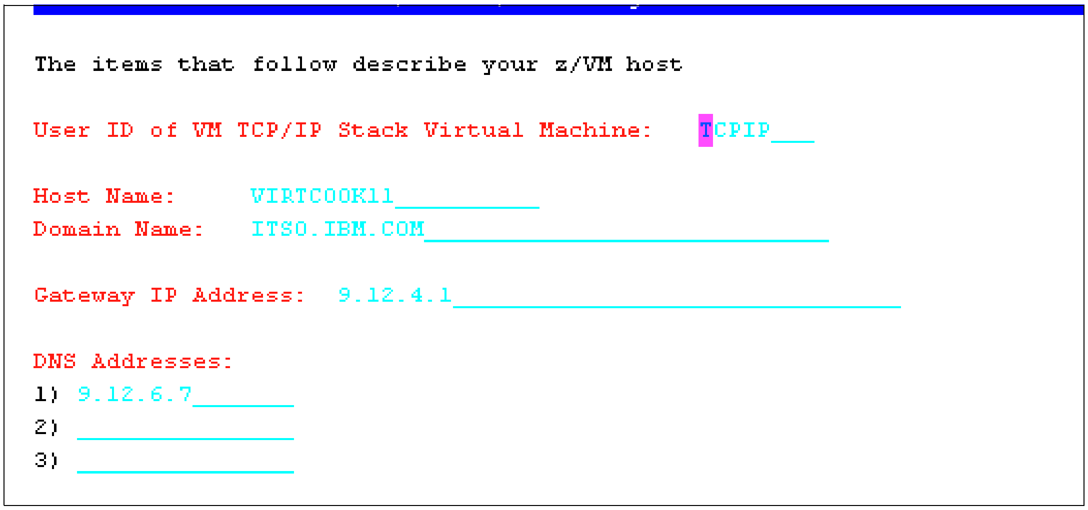


Figure 7-6 IPWIZARD panel 1

- 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 section 2.9.4, “z/VM Networking resources” on page 32 and press **F8**.

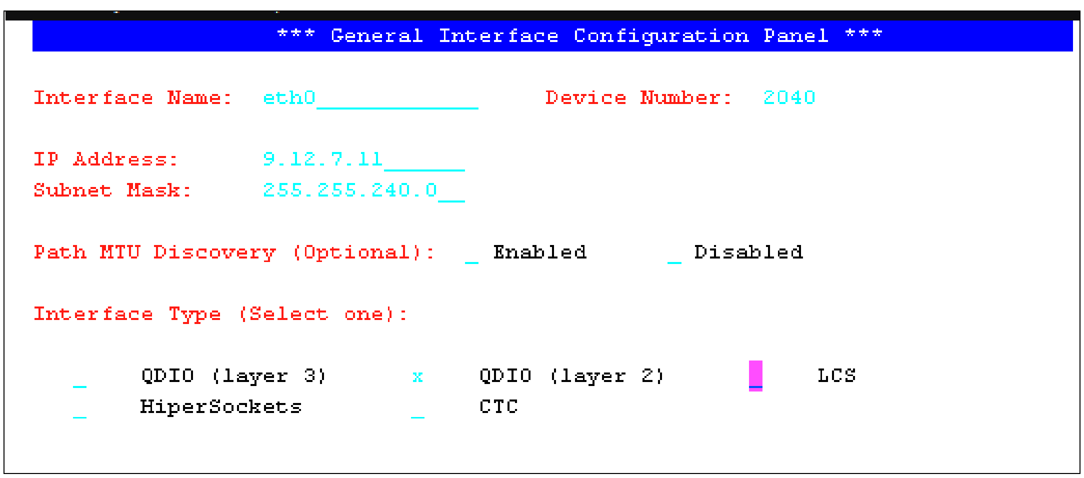


Figure 7-7 IPWIZARD panel 2

- 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**.

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

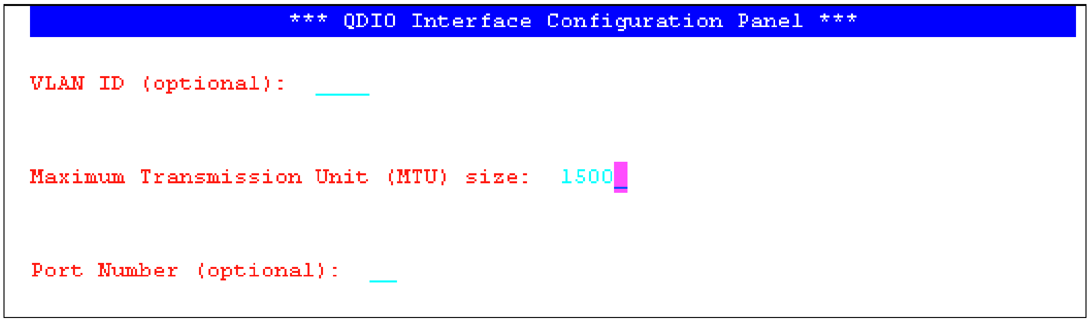


Figure 7-8 IPWIZARD panel 3

- In general, a value for the *Port Name* is no longer necessary. Press **F5** to complete the wizard:

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

- Enter **1** to restart the TCP/IP stack (you might see other warnings). Watch for the message `HCPINP8392I IPWIZARD EXEC ENDED SUCCESSFULLY`:

```
The TCP/IP stack (TCPIP) must be restarted as part of this procedure
Would you like to restart and continue?
Enter 0 (No), 1 (Yes) 1
USER DSC   LOGOFF AS  TCPIP    USERS = 11 FORCED BY MAINT
...
Successfully PINGed Interface (9.12.4.201)
Successfully PINGed Gateway (9.12.4.1)
Successfully PINGed DNS (9.12.6.7)
DTCIPW2519I Configuration complete; connectivity has been verified
DTCIPW2520I File PROFILE TCPIP created on TCPIP 198
DTCIPW2520I File TCPIP DATA created on TCPIP 592
DTCIPW2520I File SYSTEM DTCPARMS created on TCPIP 198
HCPINP8392I IPWIZARD EXEC ENDED SUCCESSFULLY
DMSVML2061I TCPIP 592 released
```

- At this point, your z/VM TCP/IP stack should be up. You should now be able to ping it from another system. If the **`IPWIZARD`** fails, you must continue debugging it until it succeeds. Double check all values. Verify that the Internet Protocol network and OSA information you were given are properly associated.

The z/VM LPAR should now be in the network.

HMC Integrated 3270 Console or 3270 emulator? At this point, your LPAR should be accessible over the network. It is recommended to ditch the Integrated 3270 panel and access your new systems with a 3270 emulator. See 3.3, "3270 emulators" on page 42 for some brief words on that subject.

To switch to a 3270 emulator, **LOGOFF** of MAINT from the Integrated 3270 Console, but you could also **DISCONNECT**. If you log off, the session is ended. It is analogous to shutting and powering down a PC. If you disconnect, your session remains where it is and is resumed when you log back on. It is analogous to turning off a PC's monitor. In general, you should **LOGOFF** of system administration virtual machines such as MAINT. However, you should always **DISCONNECT** from z/VM service machines such as TCPIP and virtual machines running Linux. Logging off from them terminates the service or crashes Linux.

7.3 Configure the XEDIT PROFILE

The **XEDIT** command looks for the file XEDIT PROFILE configuration file when it is invoked. Not all CMS virtual machines have a copy of this file, so XEDIT sessions look and behave differently. The MAINT 191 (A) disk has a PROFILE XEDIT so when you are editing files on MAINT, the values in this profile are usually in effect.

If you have never used XEDIT before, there is a cheat sheet in "XEDIT cheat sheet" on page 509. The z/VM 6.3 PDF library is on the web at the following site:

http://www-03.ibm.com/systems/z/os/zos/bkserv/zvmpdf/#zvm62

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

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

To configure the XEDIT profile on the LPAR, perform the following steps:

- Log on to MAINT on the LPAR if you are not already.
- One default setting that can be dangerous, especially if you use F12 to retrieve commands, is that PF12 is set to the **FILE** subcommand. Sometimes you might 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
  ```

  Before:

  ```
  SET PF12 FILE
  ```

  After:

  ```
  SET PF12 ?
  ```

- Save your changes with the **FILE** subcommand.
- Make the modified file available to other virtual machines by copying it to the MAINT 19E disk with file mode suffix 2:
  - Release the current 19E disk:

    ```
    ==> rel 19e
    ```

  - Link to the MAINT 19E disk read/write:

    ```
    ==> link * 19e 19e mr
    ```

```
DASD 019E LINKED R/W; R/O BY    10 USERS
```

– Access the disk as file mode F:

```
==> acc 19e f
```

– Copy it to the MAINT 19E disk (F) with file mode suffix 2 (because the MAINT 19E disk is commonly accessed with a file mode suffix of 2, files will not be seen by other virtual machines unless they have this file mode suffix):

```
==> copy profile xedit a = = f2
```

– Save the CMS named saved segment:

```
==> acc 193 g
==> sampnss cms
HCPNSD440I The Named Saved System (NSS) CMS was successfully defined in fileid 0
029.
==> ipl 190 parm savesys cms
HCPNSS440I Named Saved System (NSS) CMS was successfully saved in fileid 0029.
```

7.4 Customize the SYSTEM CONFIG file

The first configuration file read when z/VM IPLs is the SYSTEM CONFIG file.

The following changes are recommended:

- Increase retrieve key capacity
- Allow virtual disks (VDISKs) to be created
- Turn off the Disconnect Timeout feature (this will prevent idle disconnected users from being forced off the system)
- Define virtual switches to be used for Linux networking

To make these changes, perform the following steps:

- Access the PMAINT CF0 disk read/write. Use the **LINK** command with multi-read (**MR**) parameter:

```
==> link pmaint cf0 cf0 mr
```

- Use the **ACCESS** command to get access to your F disk:

```
==> acc cf0 f
```

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

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

- Edit the original file:

```
==> x system config f
====> /features
```

- 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 20 - **99**.
  – Set the Disconnect_Timeout to **off** so disconnected users do not get forced off.
  – Allow unlimited virtual disks 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
```

- The `VMLAN MACPREFIX` statement will set the first 3 bytes of the MAC address created for each virtual NIC. If you have multiple z/VM systems, increment this value to avoid having identical MAC addresses created. In this example, ***02000D*** is used because lower values have already been assigned. The `VMLAN TRANSIENT 0` statement prevents dynamic definition of Guest LANs by class G users.

```
====> bot
====> a 2
  vmlan macprefix 02000d
  vmlan limit transient 0
```

- Define two Layer 2 virtual switches with the `DEFINE VSWITCH` statements. Modify the two starting addresses of the OSA triplets to those that you specified in 2.9.4, "z/VM Networking resources" on page 32:

```
====> bot
====> a 3
/* Define layer 2 VSWITCHes VSW1 and VSW2 */
define vswitch vsw1 rdev 2103 2120 ethernet
define vswitch vsw2 ethernet
```

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

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

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

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

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

- Release and detach the `PMAINT CF0` disk with the **RELEASE** command:

```
==> rel f (det
DASD 0CF0 DETACHED
```

The `SYSTEM CONFIG` file should now be initially configured.

7.5 Configure additional network resources

The following changes are recommended to the system:

- "Turn on the z/VM FTP server" on page 132
- "Shut down and re-IPL the LPAR" on page 132
- "Test changes" on page 133

The main TCP/IP configuration file is `PROFILE TCPIP` file is on the `TCPMAINT 198` disk, which is accessed as the `D` disk.

7.5.1 Turn on the z/VM FTP server

Turn on the FTP server by performing the following steps:

- Log on to `TCPMAINT`.
- Make a backup copy of the TCP/IP configuration file, `PROFILE TCPIP D`:

  ```
  ==> copy profile tcpip d = tcpiorig = (oldd
  ```

- Edit the TCP/IP configuration file:

  ```
  ==> x profile tcpip d
  ```

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

  Before:

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

  After:

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

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

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

7.5.2 Shut down and re-IPL the LPAR

You can watch the z/VM member shut down and re-IPL from the *Integrated 3270 Console*. If you issue this command from a 3270 emulator, you will lose your session and will not see most of the shutdown process. To shut down and re-IPL the LPAR, perform the following steps:

- From the HMC, start an *Integrated 3270 Console* session for your LPAR.
- Issue the **SHUTDOWN REIPL** command:

```
==> shutdown reipl
...
```

- When z/VM comes up again, logon to MAINT.
- By default, the TCP/IP service virtual machine is not logged on (that will be customized later). Start TCP/IP with the **XAUTOLOG** command:

```
==> xautolog tcpip
Command accepted
AUTO LOGON  ***       EREP     USERS = 12
HCPCLS6056I XAUTOLOG information for TCPIP: The IPL command is verified by the I
PL command processor.
```

- Try starting a 3270 emulator session to your LPAR. You should see a logon panel. If not, you will have to debug the problem from the *Integrated 3270 Console* session. For example, you could FORCE TCPIP and logon to TCP/IP interactively and watch for error messages.

7.5.3 Test changes

To test the changes you made, perform the following steps:

- Start a 3270 emulator session.
- Log on as MAINT.
- Use the **QUERY RETRIEVE** and **QUERY VDISK** commands to see the changes that are 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
```

- Check if virtual switches are defined:

```
==> q vswitch
VSWITCH SYSTEM VSW1     Type: QDIO    Connected: 0    Maxconn: INFINITE
  PERSISTENT  RESTRICTED    ETHERNET                  Accounting: OFF
  USERBASED
  VLAN Unaware
  MAC address: 02-00-0D-00-00-01    MAC Protection: Unspecified
  IPTimeout: 5         QueueStorage: 8
  Isolation Status: OFF        VEPA Status: OFF
 Uplink Port:
  State: Ready
  PMTUD setting: EXTERNAL   PMTUD value: 8992
  RDEV: 2103.P00 VDEV: 0600 Controller: DTCVSW1  ACTIVE
  RDEV: 2120.P00 VDEV: 0600 Controller: DTCVSW2  BACKUP

VSWITCH SYSTEM VSW2     Type: QDIO    Connected: 0    Maxconn: INFINITE
  PERSISTENT  RESTRICTED    ETHERNET                  Accounting: OFF
  USERBASED
  VLAN Unaware
  MAC address: 02-00-0D-00-00-02    MAC Protection: Unspecified
```

```
  IPTimeout: 5        QueueStorage: 8
  Isolation Status: OFF        VEPA Status: OFF
 Uplink Port:
  State: Defined
  PMTUD setting: EXTERNAL   PMTUD value: 65535
```

- Try starting an FTP session to z/VM. You should get a logon prompt.

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

7.6 Add page and perm volumes

Each z/VM 6.3 LPAR is installed with one paging volume and one spool volume, being either 3390-3s or 3390-9s, depending on which type of disks the LPAR was installed onto. One spool volume per member is probably adequate for Linux needs, however, more paging volumes are recommended.

Having adequate paging space will give you *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 virtual machines combined. A second rule of thumb is to never allow your z/VM system's paging space to go above 50% used.

7.6.1 Format volumes for page space

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

Therefore, a REXX EXEC named **CPFORMAT** has been provided to allow you to format many volumes with a single command. The EXEC is in Appendix B, "Additional material" on page 513 in section "The CPFORMAT EXEC" on page 514. 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 as the real device address (using **ATTACH *realDev* ***).

> **Note:** This EXEC labels the volumes according to the convention described in section 2.3.1, "Volume labeling convention" on page 15. 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.

7.6.2 Copy the CPFORMAT EXEC to the LPAR

Perform the following steps:

- Log off from `MAINT` on the LPAR so you will be able to get the `MAINT 191` disk in read/write mode using FTP.
- Start an SSH session to the PC NFS server and **cd** to the `/var/nfs/SG248147/vm/maint/` directory, which was created when you extracted the files associated with this book:

```
# cd /var/nfs/SG248147/vm/maint
```

- List the files for the MAINT 191 disk:

```
# ls
callsm1.exec  cpformat.exec  ssicmd.exec
```

- Start an FTP session to the LPAR as MAINT. If you get a reply from the FTP server, it shows that it is configured correctly. Issue the **MPUT** subcommand to copy all files:

```
# ftp 9.12.4.201
...
Name (9.12.4.201:root): maint
331 Send password please.
Password:
230 MAINT logged in; working directory = MAINT 191
Remote system type is VM.
ftp> mput *
mput callsm1.exec [anpqy?]? a
Prompting off for duration of mput.
...
ftp> quit
```

  You should now have the **CPFORMAT EXEC**, and all other necessary files on the MAINT 191 disk.

7.6.3 Use the CPFORMAT EXEC

To use the **CPFORMAT** EXEC, perform the following steps:

- **Log in to MAINT**. You should now have access to the CPFORMAT EXEC. Edit the file to set the first character that will be used in labels. Look for the variable **firstChar**, which defaults to **"**J**"**. It is recommended that you choose a unique character for each LPAR. In this example, the **firstChar** variable was set to **"**K**"**.

```
==> x cpformat exec
/***********************************************************/
...
  Address COMMAND
  firstChar = 'K'
...
```

- 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 K<t><xxxx> where:
    <t> is type - P (page), M (perm), S (spool) or T (Temp disk)
    <xxxx> is the 4 digit address

Syntax is:
                <----------------<
   >>--CPFORMAT--.-vdev--------.--AS---.-PERM-.---------><
                 '-vdev1-vdev2-'       '-PAGE-'
                                       '-SPOL-'
                                       '-TEMP-'
```

  The following example shows how to attach one 3390-9 volume and use **CPFORMAT** to format it as paging space. Refer to the planning worksheets that should be filled out in section 2.9.5, "z/VM DASD worksheet" on page 32.

- The DASD that will be used as a paging volume on the LPAR in this example is at real device address ***136E***. Query the device to see its status:

```
==> q 136E
DASD 136E NW136E
```

- Attach the device to `MAINT` using the **ATTACH** command. This example uses the last parameter of *, which means the current virtual machine:

```
==> att 136E *
DASD 136E ATTACHED TO MAINT 136E WITH DEVCTL HYPERPAV BASE
```

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

```
==> cpformat 136E as page
Format the following DASD:
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    136E MAINT    136E 3390  NW136E 136E          0      10017

WARNING - this will destroy data!
Are you sure you want to format the DASD as PAGE space (y/n)? y
...
DASD status after:
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    136E MAINT    136E 3390  KP136E 136E          0      10017
```

This formatting job should run for a number of minutes depending on many factors.

7.6.4 Format DASD for minidisks

In addition to CP disks such as page space, system disks will be needed to create minidisks for the virtual machines. In this section, DASDs which will be used for virtual machine minidisks will be formatted:

- Start a 3270 session as `MAINT`.
- Query the DASDs that will be used for minidisks. In this example, the DASDs have real device addresses ***136b*** and ***136f***:

```
==> q 136b 136f
DASD 136B NW136B  , DASD 136F NW136F
```

- Attach the volume:

```
==> att 136b 136f *
DASD 136B ATTACHED TO MAINT 136B WITH DEVCTL HYPERPAV BASE
DASD 136F ATTACHED TO MAINT 136F WITH DEVCTL HYPERPAV BASE
```

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

```
==> cpformat 136b 136f as perm
Format the following DASD:
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    136B MAINT    136B 3390  NW136B 136B          0      10017
MAINT    136F MAINT    136F 3390  NW136F 136F          0      10017

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    136B MAINT    136B 3390  KM136B 136B          0      10017
MAINT    136F MAINT    136F 3390  KM136F 136F          0      10017
```

You should now have two volumes that can be used for minidisks. The labels are prefixed with **KM** in this example.

7.6.5 Update the SYSTEM CONFIG file

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

- Log on to MAINT.
- Link as read/write and access the PMAINT CF0 disk:

```
==> link pmaint cf0 cf0 mr
==> acc cf0 f
```

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

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

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

```
==> x system config f
====> /page &
```

Before:

```
...
/**********************************************************************/
CP_Owned   Slot 255  KP126E
...
```

After:

```
...
/**********************************************************************/
CP_Owned   Slot 254  KP136E
CP_Owned   Slot 255  KP126E
...
```

- Move down to the User_Volume_List section. User volumes (PERM) can be specified individually with the User_Volume_List statement, or with wildcards using the User_Volume_Include statement. If you are using the labeling convention enforced by the **CPFORMAT EXEC** and no other LPAR will be using the same volumes with the same prefix, add the following single line to include all PERM space as volume labels all begin with JM6.

```
====> /user_v
/**********************************************************************/
/* Shared User Volumes                                                */
/**********************************************************************/
               User_Volume_List  KV136D

/**********************************************************************/
/* User volumes for local minidisks                                   */
/**********************************************************************/

               User_Volume_List  KV126F
               User_Volume_Include KM*
...
====> file
```

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

```
User_Volume_List KM1280
User_Volume_List KM1281
User_Volume_List KM1282
...
```

- Verify the integrity of the changes with the **CPSYNTAX** command:

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

- When you have confirmed that there are no syntax errors, you can release and detach the PMAINT CF0:

```
==> rel f (det
DASD 0CF0 DETACHED
```

After the next IPL, you should have volumes formatted for paging and minidisks that are attached to the system and in use.

7.7 Configure the AUTOLOG1 PROFILE EXEC

When z/VM IPLs, normally the AUTOLOG1 virtual machine is logged on (unless the NOAUTOLOG parameter is specified at IPL). Its PROFILE EXEC is run when CMS IPLs. It is recommended that the following tasks be accomplished using this file:

- Configure Linux to shut down gracefully using the **SET SIGNAL** command
- Limit minidisk cache
- Start virtual machines that should be started using the **XAUTOLOG** command

To configure the AUTOLOG1 PROFILE EXEC, perform the following steps:

- Log on to AUTOLOG1.
- Before pressing Enter at the VM READ prompt, type **acc (noprof** so that the **PROFILE EXEC** is not run.

```
LOGON AUTOLOG1
z/VM Version 6 Release 3.0, Service Level 1301 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES:   NO RDR,   NO PRT,   NO PUN
LOGON AT 10:44:10 EDT WEDNESDAY 09/04/13
z/VM V6.3.0    2013-09-04 07:58
==> acc (noprof
```

- Make a backup copy of the PROFILE EXEC:

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

- Edit the PROFILE EXEC and add a line so the virtual machine PERFSVM is started at z/VM IPL time:

```
==> x profile exec
====> /customer
...
```

```
/* Customer processing can be added here                              */
/*********************************************************************/
"CP XAUTOLOG TCPIP"            /* Autolog TCPIP                      */
"CP SET MDC STOR OM 256M"      /* Limit minidisk cache in CSTOR      */
"CP SET SIGNAL SHUTDOWN 600"   /* Allow guests 10 min to shut down */
...
====> file
```

The `PROFILE EXEC` on `AUTOLOG1 191` disk should be configured for the LPAR.

7.7.1 Shut down and re-IPL the LPAR

It is recommended that you again shut down and re-IPL to test the changes.

- Log on to `MAINT`.
- Before you shut down, note that you have only one page volume (`JV126E` in this example) using the **`QUERY ALLOC PAGE`** command:

```
==> q alloc page
EXTENT     EXTENT  TOTAL  PAGES   HIGH    %
VOLID  RDEV      START        END  PAGES IN USE   PAGE USED
------ ---- ---------- ---------- ------ ------ ------ ----
KP126E 126E          1      10016  1761K     46     55   1%
                                  ------ ------        ----
SUMMARY                            1761K     46          1%
USABLE                             1761K     46          1%

==> shutdown reipl
...
```

- If you are using a 3270 emulator, you will lose your session. If you watch the HMC, the LPAR should immediately turn from white to green, then return to white after a minute or so.
- After the system comes back, logon as `MAINT`.
- Use the **`QUERY ALLOC PAGE`** command. You should now see that you have two page volumes:

```
==> q alloc page
                EXTENT     EXTENT  TOTAL  PAGES   HIGH    %
VOLID  RDEV      START        END  PAGES IN USE   PAGE USED
------ ---- ---------- ---------- ------ ------ ------ ----
KP136E 136E          0      10016  1761K      0      0   0%
KP126E 126E          1      10016  1761K     51     60   1%
                                  ------ ------        ----
SUMMARY                            3521K     51          1%
USABLE                             3521K     51          1%
```

The output shows that there are two paging volumes constituting 3251 K pages, or about 13 GB of page space (a page is 4 KB). This is not much page space, but is sufficient for the relatively small setup that is described in this book. You will probably want to start with much more page space than this.

7.8 Create LNXMAINT for common files

Now it is time to define your first z/VM virtual machine, `LNXMAINT`. It will be used to store files that will be shared by Linux virtual machines.

7.8.1 Define the user in the USER DIRECT file

A small 20-cylinder minidisk is allocated at virtual address 191 and a larger 500-cylinder minidisk (approximately 350 MB), to be shared by many guests, is defined at virtual address 192. Use the free DASD designated as PERM space on your worksheet (section 2.9.5, “z/VM DASD worksheet” on page 32). In this example, it is ***KM136F***. Cylinder 0 should always be reserved for the label; therefore, you should start minidisks at cylinder 1.

- Make a copy of the original USER DIRECT file:

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

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

```
==> x user direct c
====> bottom
====> a 9
...
*------------------------------------------------------------
* z/VM system virtual machines are above, user defined below
*------------------------------------------------------------
USER LNXMAINT LNXMAINT 64M 128M G          1
 INCLUDE TCPCMSU                           2
 LINK TCPMAINT 592 592 RR                  3
 MDISK 0191 3390 0001 0020 KM136F MR READ     WRITE    MULTIPLE   4
 MDISK 0192 3390 0021 0500 KM136F MR ALL      WRITE    MULTIPLE   5
*                                                                 6
====> file
```

Note the following points for the numbers in black:

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

2 Include the profile that is 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 KM136F

5 Define a 192 minidisk of size 500 cylinders (approximately 350 MB) from volume KM136F with the special read password of **ALL**, which allows read access from any virtual machine without a disk password

6 An empty comment line for better readability

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

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

File USER DISKMAP A has been created.
```

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

```
==> x user diskmap
```

```
====> prefix off
```

- Search for all other overlaps with the **ALL** subcommand:

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

You should see no overlaps. If you see any, correct your USER DIRECT.

- Type **ALL** with no argument again to get out of this mode

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

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

```
====> all /gap
                                            0         500         501    GAP

--------------------  6  line(s) not displayed --------------------
                                            0           0           1    GAP

--------------------  5  line(s) not displayed --------------------
                                            0           0           1    GAP

--------------------  345  line(s) not displayed --------------------

====> all
```

Three GAPs should be listed on the right side:

– 501 cylinders on the $$$$$$ volume
– 1 cylinder on the $$LNX1 volume
– 1 cylinder on the volume used for LNXMAINT 191 and 192 disks (KM136F in this example)

You do not have to worry about the first two gaps because they are expected given the layout of the default USER DIRECT file. To avoid a one-cylinder gap being reported on each user volume, it is recommended to use the virtual machine $ALLOC$. This user is set to NOLOG, which means it can never be logged on to. Thus, it is not a conventional virtual machine, rather, it is a convenient place to put dummy minidisk definitions for cylinder 0 of all PERM volumes.

- Get out of the file USER DISKMAP with the **QUIT** command or by pressing **F3**.
- Edit the USER DIRECT file again and add a new minidisk definition at virtual address A04 for the first cylinder of new DASD volume you added. In this example, there is just one:

```
==> x user direct
====> /user $alloc
USER $ALLOC$  NOLOG
 MDISK A00 3390 000 001 KV136C R
 MDISK A01 3390 000 001 KV136D R
 MDISK A02 3390 000 001 KV126C R
 MDISK A03 3390 000 001 KV126F R
 MDISK A04 3390 000 001 KM136F R
*
```

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

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

File USER DISKMAP A has been created.
==> x user diskmap
====> prefix off
```

```
====> all /gap
                                            0         500         501    GAP

--------------------  6  line(s) not displayed --------------------
                                            0           0           1    GAP

--------------------  351 line(s) not displayed --------------------
```

Only two gaps are now reported. They are on volumes `$$$$$$` and `$$LNX1`.

- Quit XEDIT by pressing **F3**.

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

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

```
==> directxa user
z/VM USER DIRECTORY CREATION PROGRAM - VERSION 6 RELEASE 3.0
EOJ DIRECTORY UPDATED AND ON LINE
HCPDIR494I User directory occupies 58 disk pages
```

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

You have now defined your first z/VM virtual machine, which is named `LNXMAINT`, and brought it online.

7.8.2 Format the LNXMAINT minidisks

Now you should be able to log on to the new virtual machine and format its two minidisks:

- Log off from `MAINT`.
- Log on to `LNXMAINT`.

```
LOGON LNXMAINT
z/VM Version 6 Release 3.0, Service Level 1301 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES:   NO RDR,   NO PRT,   NO PUN
LOGON AT 10:56:36 EDT WEDNESDAY 09/04/13
z/VM V6.3.0    2013-09-04 07:58

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 virtual machine, however, it has never been formatted as a CMS file system.

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

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

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

```
==> format 192 d
DMSFOR603R FORMAT will erase all files on disk D(192). Do you wish to continue?
Enter 1 (YES) or 0 (NO).
```

```
1
DMSFOR605R Enter disk label:
lxm192
DMSFOR733I Formatting disk D
DMSFOR732I 500 cylinders formatted on D(192)
```

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

```
==> q disk
LABEL  VDEV M  STAT   CYL TYPE BLKSZ   FILES  BLKS USED-(%) BLKS LEFT  BLK TOTAL
LXM191 191  A   R/W    20 3390 4096        0          7-00       3593       3600
LXM192 192  D   R/W   500 3390 4096        0         13-00      89987      90000
MNT190 190  S   R/O   207 3390 4096      696      18085-49      19175      37260
MNT19E 19E  Y/S R/O   500 3390 4096     1124      30404-34      59596      90000
```

7.8.3 Create a PROFILE EXEC

Create a simple **`PROFILE EXEC`** that is run each time that this virtual machine is logged on.

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

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

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

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

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

- By default, CMS tries to access the `191` disk as `A` and the `192` disk as `D`. Also, you should have the `TCPMAINT 592` disk accessed as `E`. Verify these three disks are accessed with the **`QUERY DISK`** command:

```
==> q disk
LABEL  VDEV M  STAT   CYL TYPE BLKSZ   FILES  BLKS USED-(%) BLKS LEFT  BLK TOTAL
LXM191 191  A   R/W    20 3390 4096        1          8-01       3592       3600
LXM192 192  D   R/W   500 3390 4096        0         13-00      89987      90000
TCM592 592  E   R/O   140 3390 4096      858      10312-41      14888      25200
MNT190 190  S   R/O   207 3390 4096      696      18085-49      19175      37260
MNT19E 19E  Y/S R/O   500 3390 4096     1124      30404-34      59596      90000
```

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

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

7.8.4 Copy files associated with this book

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

- **Log off from LNXMAINT** so that the 192 disk can be accessed read/write.
- **Start an SSH session on the NFS server** and change the directory to the VM files associated with this book:

```
# cd /var/nfs/SG248147/vm/lnxmaint
```

- List the files for the LNXMAINT 192 disk:

```
# ls
profile.exec  sample.conf-rh6  sample.parm-s11  swapgen.exec
rhel64.exec   sample.parm-rh6  sles11s3.exec
```

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

```
# ftp 9.12.4.201
Connected to 9.12.4.201 (9.12.4.201).
220-FTPSERVE IBM VM Level 630 at VIRTCOOK13.ITSO.IBM.COM, 11:01:22 EDT WEDNESDAY
2013-09-04
220 Connection will close if idle for more than 5 minutes.
Name (9.12.4.201:root): lnxmaint
Password:
230 LNXMAINT logged in; working directory = LNXMAINT 191
Remote system type is VM.
ftp> cd lnxmaint.192
250 Working directory is LNXMAINT 192
ftp> prompt
Interactive mode off
ftp> mput *
...
ftp> quit
```

- Log on to LNXMAINT.
- Use the **FILELIST** command to show the files on the D disk:

```
==> filel * * d
 LNXMAINT FILELIST A0  V 169  Trunc=169 Size=7 Line=1 Col=1 Alt=0
Cmd   Filename Filetype Fm Format Lrecl    Records     Blocks   Date     Time
      PROFILE  EXEC     D1 V         63         17          1  9/04/13 11:02:19
      RHEL64   EXEC     D1 V         63         14          1  9/04/13 11:02:19
      SAMPLE   CONF-RH6 D1 V         38         13          1  9/04/13 11:02:19
      SAMPLE   PARM-RH6 D1 V         80          3          1  9/04/13 11:02:19
      SAMPLE   PARM-S11 D1 V         69         11          1  9/04/13 11:02:19
      SLES11S3 EXEC     D1 V         63         12          1  9/04/13 11:02:19
      SWAPGEN  EXEC     D1 V         72        485          6  9/04/13 11:02:19
```

- Logoff of LNXMAINT.

You should now have copied the files associated with this book to LNXMAINT 192.

7.9 Create user LNXADMIN for Linux administration

Now it is time to create the virtual machine, which will serve a number of administrative purposes:

Note: This chapter shows how to prepare LNXADMIN for RHEL 6.4 usage as an example. If you plan to use SLES, you have to adapt accordingly.

- The Linux installation server: A file system *tree* of RPMs and other files that are required for installation are made available with NFS.
- The clone server: For cloning from the golden image to target virtual machines (see Chapter 10, "Configure RHEL 6.4 for cloning" on page 193).
- The Red Hat kickstart server: For hosting the files necessary for automated installations (see Chapter 22, "DirMaint, SMAPI, and RACF" on page 371).
- The administration server for other systems management tools such as xCAT.

To create this virtual machine, perform the following steps:

- Log on to `MAINT`.
- Determine the number of logical processors active with the **`QUERY PROCESSORS`** command:

```
==> q proc
PROCESSOR 00 MASTER CP
PROCESSOR 01 ALTERNATE CP
PROCESSOR 02 PARKED CP
PROCESSOR 03 PARKED CP
PROCESSOR 04 PARKED CP
PROCESSOR 05 PARKED CP
PROCESSOR 06 STANDBY CP
PROCESSOR 07 STANDBY CP
PROCESSOR 08 STANDBY CP
PROCESSOR 09 STANDBY CP
```

  Ensure that you have at least two logical processors. If not, you will have to change the number of virtual processors defined for virtual servers. There is no point in defining more virtual processors in a virtual server than there are logical processors available to z/VM.

- Make a backup of the `USER DIRECT` file:

```
==> copy user direct c = direwrks = (rep
```

- Edit the `USER DIRECT` file:

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

  In the `USER DIRECT` file, you can group statements that will be common to many user definitions in a construct called a *profile*. This profile can then become part of the user definitions using the `INCLUDE` statement. You used the existing profile `TCPCMSU` when you defined the `LNXMAINT` user. Next, you will create a user directory profile for Linux systems.

- 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, type the **`""`** prefix commands to duplicate the `IBMDFLT` profile on lines 38 - 50:

```
"" 38 *
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 *
"" 50 *****************************
```

- Press **Enter** and the block will be duplicated.
- Edit the duplicated profile by deleting the two `LINK MAINT 040x` lines, and inserting the lines that are shown in bold text:

```
PROFILE LNXDFLT
  COMMAND SET VSWITCH VSW1 GRANT &USERID
  COMMAND DEFINE NIC 600 TYPE QDIO
  COMMAND COUPLE 600 TO SYSTEM VSW1
  COMMAND SET VSWITCH VSW2 GRANT &USERID
  COMMAND DEFINE NIC 700 TYPE QDIO
  COMMAND COUPLE 700 TO SYSTEM VSW2
  CPU 00 BASE
  CPU 01
  IPL CMS
  MACHINE ESA 8
  IUCV ALLOW
  SPOOL 000C 2540 READER *
  SPOOL 000D 2540 PUNCH A
  SPOOL 000E 1403 A
  CONSOLE 0009 3215 T
  LINK MAINT 0190 0190 RR
  LINK MAINT 019D 019D RR
  LINK MAINT 019E 019E RR
  LINK LNXMAINT 0192 0191 RR
  LINK TCPMAINT 0592 0592 RR
```

Notes regarding the preceding lines:

- The six **COMMAND** lines give the virtual machine access to virtual switches `VSW1` and `VSW2` at logon time when the virtual machine is created. This precludes the need to add **VSWITCH GRANT** statements each time a Linux virtual machine is created.
- The two **CPU** lines define two virtual CPUs. It is recommended to set the number of virtual CPUs less than or equal the number of physical CPUs.
- The `MACHINE` statement sets the virtual machine type to ESA with a maximum of 8 CPUs. Even if your hardware does not have 8 IFLs, it is alright to set the maximum to 8 to leave *headroom*.
- The `MACHINE` statement sets the virtual machine type to ESA with a maximum of 8 CPUs. Even if your hardware does not have 8 IFLs, it is alright to set the maximum to 8 to leave *headroom*.
- The **IUCV ALLOW** line allows virtual machines to connect to other virtual machines, such as the Linux Terminal Server, by using IUCV.
- The second to last line provides read access to `LNXMAINT 192` disk as the user's `191` disk.

- Go to the bottom of the file and add the definition for a new user named LNXADMIN. This virtual machine is given privilege class B, aside from the typical class G, in order to run the **FLASHCOPY** command:

```
USER LNXADMIN LNX4VM 256M 1G BG
 INCLUDE LNXDFLT
 OPTION LNKNOPAS
 MDISK 0100 3390 0001 10016 KM136B MR LNX4VM LNX4VM LNX4VM
 MDISK 0101 3390 0521  9496 KM136F MR LNX4VM LNX4VM LNX4VM
```

This virtual machine will have the following minidisks and virtual disks (see Table 7-1).

Table 7-1 Minidisks to be defined

Minidisk or virtual disk	Description
LNXADMIN 100	The root file system of the Linux administration system. This will serve as the administration point for all your Linux virtual servers.
LNXADMIN 101	Minidisk used to create a logical volume mounted over /var/. This file system is used to make the installation trees, files associated with this book, and possibly other data available over NFS.
300-301	These are virtual disk 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 virtual disks are created.

- Go back to the top of the file and search for string USER $ALLOC$. Add cylinder 0 of each of the new volumes to this dummy user ID so they do not show up as gaps in the USER DISKMAP report file. In this case, there is only one new volume:

```
====> top
====> /user $alloc$
USER $ALLOC$  NOLOG
 MDISK A00 3390 000 001 KV136C R
 MDISK A01 3390 000 001 KV136D R
 MDISK A02 3390 000 001 KV126C R
 MDISK A03 3390 000 001 KV126F R
 MDISK A04 3390 000 001 KM136F R
 MDISK A05 3390 000 001 KM136B R
...
====> file
```

- Run **DISKMAP** to check for overlaps and gaps:

```
==> diskmap user
==> x user diskmap
====> all /gap/|/overlap/
...
====> quit
```

- When the disk layout is correct, run the **DIRECTXA** command to bring the changes online:

```
==> directxa user
z/VM USER DIRECTORY CREATION PROGRAM - VERSION 6 RELEASE 3.0
EOJ DIRECTORY UPDATED AND ON LINE
HCPDIR494I User directory occupies 58 disk pages

EOJ DIRECTORY UPDATED AND ON LINE
```

You have now defined the virtual machine that will be the Linux administrative system.

7.9.1 Set LNXADMIN to start an IPL time

It is recommended that the new Linux administrative system be started at IPL time. To do so, perform the following tasks:

- Use the **LINK** and **ACCESS** commands to link and access the `AUTOLOG1 191` disk read/write. This is the disk with the common EXEC that is run at IPL time for each member:

```
==> link autolog1 191 1191 mr
==> acc 1191 f
```

- Edit the file **PROFILE EXEC**. Add a line to automatically start the `LNXADMIN` identity with the **XAUTOLOG** command:

```
==> x profile exec f   // add one line
...
"CP XAUTOLOG TCPIP"             /* Autolog TCPIP                   */
"CP SET MDC STOR OM 256M"       /* Limit minidisk cache in CSTOR    */
"CP SET SIGNAL SHUTDOWN 600"    /* Allow guests 10 min to shut down */
"CP XAUTOLOG LNXADMIN"          /* Start the Linux admin machine    */
...
====> file
```

- Release and detach the `AUTOLOG1 191` disk with the **RELEASE** command:

```
==> rel f (det
DASD 1191 DETACHED
```

The `LNXADMIN` user will now be automatically started.

Refer to sections 5.12, “z/VM security issues” on page 99 and 5.13, “Back up and restore your z/VM system” on page 100.

Part 2

Red Hat Enterprise Linux 6.4

This part of the book focuses on Red Hat Enterprise Linux (RHEL). It consists of the following chapters:

- Chapter 8, "Install Red Hat Enterprise Linux on LNXADMIN" on page 151, describes how to install and configure RHEL 6.4 onto the *Linux Administration* server, which does the cloning and other tasks.
- Chapter 9, "Installing and configuring the RHEL 6.4 golden image" on page 175, describes how to install and configure the *golden image* which it is cloned from.
- Chapter 10, "Configure RHEL 6.4 for cloning" on page 193, explains how to prepare z/VM virtual machines and clone your first virtual server both manually and by using a shell script.
- Chapter 11, "Create RHEL 6.4 appliances" on page 205, describes how to create virtual appliances from cloned RHEL 6.4 servers.
- Chapter 12, "Installing Red Hat Enterprise Linux with kickstart" on page 229, describes how to use Red Hat's **kickstart** tool to create Linux systems. This is fundamentally different from cloning in that an automated installation is implemented. You can try kickstart and you can also try cloning. Understand that they try to accomplish the same goal of being able to quickly get Linux systems up and running, and that you do not need to use both.
- Chapter 13, "Service Linux with the Red Hat Network" on page 235, describes how the *Red Hat Network* works. It provides centralized management and provisioning for multiple RHEL 6.4 systems.
- Chapter 14, "Red Hat Network Satellite Server" on page 241, describes Red Hat Network Satellite Server, which is an easy-to-use, advanced systems management platform for your Linux infrastructure. It shows how to install RHN Satellite, how to service Linux, and how to provision your Linux guests easily.

Kickstart is a very easy and fast way to provision you Linux guests in any supported Linux platform. It re-creates the operating system (OS) from scratch by using the kickstart profile configuration file that installs the new OS unintended and sets up the new guest according to what was predefined in the kickstart file.

Usually, Linux administration is done by the same team that manages Linux on all platforms. By using kickstart, you can create a basic profile that can be used in all supported platforms and customize Linux profiles as needed.

Cloning is another technique to provision Linux guests. This requires a better understanding of the z/VM environment and z/VM skills. It is a very fast process if the client has the FLASHCOPY feature enabled. It basically clones the discs from a golden image to new discs that will be used by the new Linux guest. The process can be automated using the cloning scripts supplied with this book.

8

Install Red Hat Enterprise Linux on LNXADMIN

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

— Albert Einstein

By now, you have created a new z/VM *user ID* or Single Configuration Virtual Machine (SCVM), `LNXMAINT`. Its main purpose is to provide a common CMS disk for all Linux virtual machines.

It is time to create the first *Identity* or Multi-Configuration Virtual Machine (MCVM), `LNXADMIN`. An MCVM can be logged on to all members of the single system image (SSI) at the same time. Therefore, it is not possible to migrate an MCVM between SSI members.

This virtual machine serves a number of administrative purposes:

- The Linux installation server: A file system *tree* of RPMs and other files that are required for installation are made available with NFS.
- The clone server: For cloning from the golden image to target virtual machines (see Chapter 10, "Configure RHEL 6.4 for cloning" on page 193).
- The Red Hat kickstart server: For hosting the files necessary for automated installations (see Chapter 12, "Installing Red Hat Enterprise Linux with kickstart" on page 229).
- The administration server for other systems management tools such as xCAT.

In this section, you perform the following tasks:

- "Install the Linux administration system" on page 151
- "Configure the Linux administration system" on page 164

8.1 Install the Linux administration system

In this section, you install RHEL 6.4 on to the `IDENTITY LNXADMIN`.

8.1.1 Prepare RHEL 6.4 bootstrap files

To IPL an RHEL 6.4 installation system, five bootstrap files must be prepared. Three are punched to z/VM reader and then IPLed: 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 and the fifth file is the `redhat.exec`, a small REXX EXEC that is commonly used to clean out the reader, punch the three files, and IPL the reader.

Think of these as the files that are on as a PC Linux boot CD or DVD. Also, a sample RHEL 6.4 parameter file, configuration file, and installation EXEC are supplied and should be on the `LNXMAINT 192` disk (see section 5.10.4, "Copy files associated with this book" on page 95). Therefore, only the kernel and RAMdisk need to be copied.

Note: Summary of the installation process:

Copy the following files to the *user ID* that you will install the RHEL guest using FTP:

- initrd.img, kernel.img, sample.conf-rh64, sample.parm-rh64, and rhel64.exec
- Edit with the correct values the files: sample.conf-rh64, sample.parm-rh64
- Start the installation process with rhel64.exec

1. Start a Secure Shell (SSH) session as `root` on the NFS server.
2. List the contents of the `/srv/nfs/rhel64/` directory. The ISO image should be mounted over it:

```
# ls /srv/nfs/rhel64
boot.cat     Packages                RELEASE-NOTES-mr-IN.html
EULA         README                  RELEASE-NOTES-or-IN.html
...
```

   If you do not see the distribution files, refer back to section 4.3, "Set up an NFS server" on page 49.

3. Use the **ftp** command to copy the RHEL 6.4 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 /srv/nfs/rhel64/images
# ftp 9.12.7.11
Name (9.12.7.11: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 rhel64.initrd
...
ftp> put kernel.img rhel64.kernel
...
ftp> quit
```

4. From your 3270 session, log off from `MAINT` if you are still logged on.
5. Log on to `LNXMAINT`.

6. The files SAMPLE PARM-RH6, and SAMPLE CONF-RH6 should exist on the LNXMAINT 192 (D) disk if they were copied there. If they were not copied yet, see section 5.10.4, "Copy files associated with this book" on page 95 to copy them accordingly. Use the **FILELIST** command to verify that the files were copied:

```
==> filel sample *-rh6 d
LNXMAINT FILELIST A0  V 169  Trunc=169 Size=2 Line=1 Col=1 Alt=0
Cmd   Filename Filetype Fm Format Lrecl    Records     Blocks   Date     Time
      SAMPLE   PARM-RH6 D1 V         80          3          1  7/15/13 13:42:59
      SAMPLE   CONF-RH6 D1 V         38         13          1  7/15/13 13:42:58
```

7. The RHEL64 EXEC should exist as well as the RHEL 6.4 kernel and initial RAMdisk. Verify that they were in fixed 80-byte record format. Use the **FILELIST** command to verify that the files were copied:

```
==> filel rhel64 * d
LNXMAINT FILELIST A0  V 169  Trunc=169 Size=3 Line=1 Col=1 Alt=0
Cmd   Filename Filetype Fm Format Lrecl    Records     Blocks   Date     Time
      RHEL64   KERNEL   D1 F         80     113594       1719  7/24/13 12:45:16
      RHEL64   INITRD   D1 F         80     228920       4472  7/24/13 12:45:07
      RHEL64   EXEC     D1 V         63         11          1  7/15/13 13:42:57
```

8. Verify that the **RHEL64 EXEC** file has the correct information. Note the kernel and RAMdisk have hardcoded file names (RHEL6), but the file name of the parameter file will be the user ID (**userid()** function) of the user running the EXEC:

```
==> type rhel64 exec d
...
Address 'COMMAND'
'CP SPOOL PUN *'
'CP CLOSE RDR'
'CP PURGE RDR ALL'
'PUNCH RHEL64 KERNEL * (NOHEADER'
'PUNCH' Userid() 'PARM-RH6 * (NOHEADER'
'PUNCH RHEL64 INITRD * (NOHEADER'
'CP CHANGE RDR ALL KEEP'
'CP IPL 00C CLEAR'
Exit
```

9. There are two text files needed to install RHEL 6.4: A parameter file and a configuration file. A sample parameter file is provided, named SAMPLE PARM-RH6. It has some values. The most important value, the CMSCONFFILE variable, points to the configuration file, which remains on a CMS minidisk. Copy the sample parameter file to a new file with a file name of LNXADMIN. Change the configuration file variable to point to a file with the same file name:

```
==> copy sample parm-rh6 d lnxadmin = =
==> x lnxadmin parm-rh6 d
root=/dev/ram0 ro ip=off ramdisk_size=40000
CMSDASD=191 CMSCONFFILE=LNXADMIN.CONF-RH6
vnc vncpassword=12345678
```

10. Copy the sample configuration file and modify the appropriate fields. Refer to the worksheet in section 2.9.7, "Host names and IP addresses worksheet" on page 33. The following values are used for the example in this book:

```
==> copy sample conf-rh6 d lnxadmin = =
==> x lnxadmin conf-rh6
DASD=100-101,300-301
HOSTNAME=virtcook7.itso.ibm.com
NETTYPE=qeth
```

```
IPADDR=9.12.7.7
SUBCHANNELS=0.0.0600,0.0.0601,0.0.0602
NETMASK=255.255.240.0
SEARCHDNS=itso.ibm.com
GATEWAY=9.12.4.1
DNS=9.12.6.7
MTU=1500
PORTNAME=DONTCARE
PORTNO=0
LAYER2=1
VSWITCH=1
```

11. Linux user IDs pick up their `PROFILE EXEC` from `LNXMAINT 192`. This file runs when you press **Enter** at the `VM READ` prompt. It creates two virtual disks with the `SWAPGEN EXEC` to later be used as swap spaces. It also performs a few other functions, including IPLing Linux automatically if the virtual machine is logged on disconnected. View the contents of the `PROFILE EXEC` with 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 300 524288' /* create a 256M VDISK disk swap space */
'SWAPGEN 301 1048576' /* create a 512M VDISK disk swap space */
'PIPE CP QUERY' userid() '| var user'
parse value user with id . dsc .
if (dsc = 'DSC') then /* user is disconnected */
  'CP IPL 100'
else  /* user is interactive -> prompt */
 do
  say 'Do you want to IPL Linux from minidisk 100? y/n'
  parse upper pull answer .
  if (answer = 'Y') then 'CP IPL 100'
 end
```

8.1.2 Install RHEL 6.4

Perform the following steps to begin the Linux installation:

1. Log on to `LNXADMIN`. The **`PROFILE EXEC`** from the `LNXMAINT 192` disk should prompt you to IPL minidisk `100`. Since there is nothing installed yet, answer **no**:

```
LOGON LNXADMIN
00: z/VM Version 6 Release 3.0, Service Level 0000 (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 08:25:50 EDT THURSDAY 06/06/13
00: Command complete
00: NIC 0600 is created; devices 0600-0602 defined
00: NIC 0600 is connected to VSWITCH SYSTEM VSW1
z/VM V6.3.0    2013-06-04 12:50
Enter
DMSACP723I A (191) R/O
DMSACP723I C (592) R/O
DIAG swap disk defined at virtual address 300 (64988 4K pages of swap space)
DIAG swap disk defined at virtual address 301 (129980 4K pages of swap space)
Do you want to IPL Linux from minidisk 100? y/n
n
```

2. Set the memory size to 1 GB with the CP **DEFINE STORAGE** command:

```
==> def stor 1g
00: STORAGE = 1G
00: Storage cleared - system reset.
```

3. IPL CMS, and again answer **no**:

```
==> ipl cms
z/VM V6.3.0    2013-06-04 12:50

DMSACP723I A (191) R/O
DMSACP723I C (592) R/O
DIAG swap disk defined at virtual address 300 (64989 4K pages of swap space)
DIAG swap disk defined at virtual address 301 (129981 4K pages of swap space)
Do you want to IPL Linux from minidisk 100? y/n
n
Ready; T=0.01/0.03 11:37:42
```

4. To begin the installation program, run the **RHEL64 EXEC**. You should see many screens of output scrolling by:

```
==> rhel64
   00: 0000003 FILES PURGED
   00: RDR FILE 0011 SENT FROM LNXADMIN PUN WAS 0011 RECS 114K CPY  001 A NOHOLD NO
   KEEP
   00: RDR FILE 0012 SENT FROM LNXADMIN PUN WAS 0012 RECS 0003 CPY  001 A NOHOLD NO
   KEEP
   00: RDR FILE 0013 SENT FROM LNXADMIN PUN WAS 0013 RECS 229K CPY  001 A NOHOLD NO
   KEEP
   00: 0000003 FILES CHANGED
   00: 0000003 FILES CHANGED
   Initializing cgroup subsys cpuset
   Initializing cgroup subsys cpu
   Linux version 2.6.32-358.el6.s390x (mockbuild@s390-002.build.bos.redhat.com) (gc
   c version 4.4.7 20120313 (Red Hat 4.4.7-3) (GCC) ) #1 SMP Tue Jan 29 12:06:31 ES
   T 2013
   setup: Linux is running as a z/VM guest operating system in 64-bit mode
   ...
   Connect now to 9.12.7.7 and log in as user 'install' to start the installation.
   E.g. using: ssh -x install@9.12.7.7
   For VNC or text mode, disable X11 forwarding (recommended) with 'ssh -x'.
   For X11, enable X11 forwarding with 'ssh -X'.

   You may log in as the root user to start an interactive shell.
   DEBUG kernel:eth0: no IPv6 routers present
```

> **Important:** The message says to log in as the user **install**, however, there is an intermediate step. There is an issue where the Red Hat installer does not recognize disks that have been formatted with **CPFMTXA**.

5. If you have followed all the steps in this book, this will be the case (if you have previously used **dasdfmt** to format these minidisks, you can skip this step). You must first start an SSH session, log in as root, and use **dasdfmt** to format the disks.

   To do so, perform the following steps:

   a. Start an SSH session to the installation system and log in as `root`. A password will not be required.

   b. Invoke the **lsdasd** command and observe the disks:

```
# lsdasd
Bus-ID     Status      Name      Device  Type  BlkSz  Size      Blocks
==============================================================================
0.0.0100   n/f         dasdb     94:4    ECKD
0.0.0101   n/f         dasdc     94:8    ECKD
0.0.0300   active      dasdd     94:12   FBA   ???    256MB     ???
0.0.0301   active      dasde     94:16   FBA   ???    512MB     ???
```

c. Format the minidisks in parallel with the following **for** loop:

```
# for i in b c
> do
>   dasdfmt -b 4096 -y -f /dev/dasd$i &
> done
[1] 1001
[2] 1002
```

d. You might need to press **Enter** after a few minutes to see the jobs in the background complete:

```
Rereading the partition table... ok
Finished formatting the device.
Rereading the partition table... ok
Finished formatting the device.

[1]   Done                    dasdfmt -b 4096 -y -f /dev/dasd$i
[2]   Done                    dasdfmt -b 4096 -y -f /dev/dasd$i
```

e. After the **for** loop completes, exit that SSH session:

```
# exit
```

f. Start an SSH session to the installation system and log in as **install**. A password will not be required.

6. Figure 8-1 shows the initial panel 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.

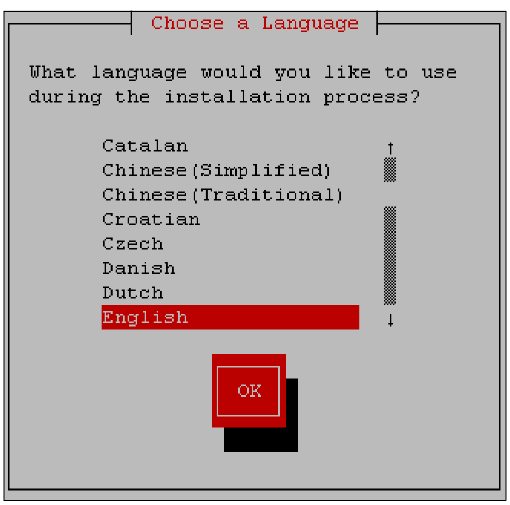


Figure 8-1 Initial panel of installer

7. The `Choose a Language` panel should appear. Select your language, Tab to **OK**, and press **Enter**.

8. The *Installation Method* panel should appear. Choose **NFS directory** for the installation method, and select **OK**.

9. The *NFS Setup* panel 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 8-2, which uses the NFS server at IP address ***9.12.5.251***.

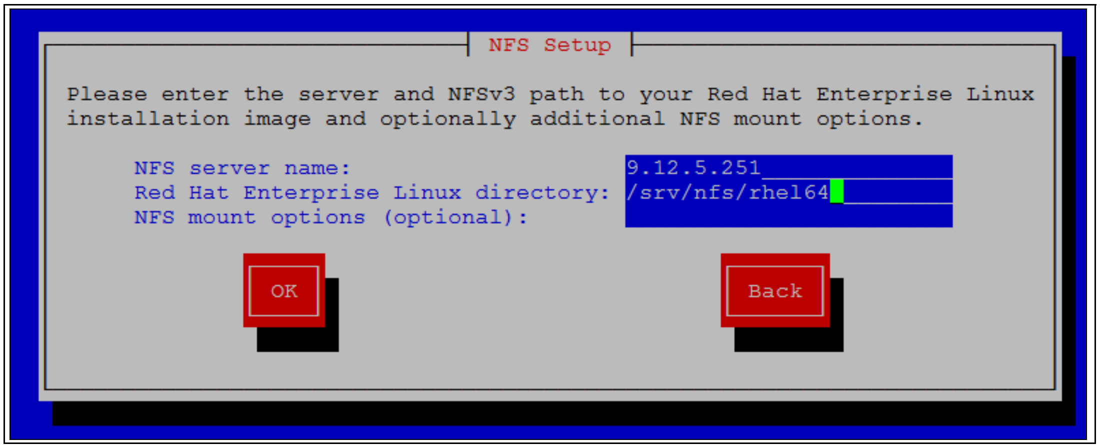


Figure 8-2 NFS setup panel

10. Now the block mode windows should end and the installation program (anaconda) should start a Virtual Network Computing (VNC) server. You should see messages similar to the following:

```
...
detecting hardware...
waiting for hardware to initialize...
detecting hardware...
waiting for hardware to initialize...
Running anaconda 13.21.195, the Red Hat Enterprise Linux system installer - please
wait.
17:06:12 Starting VNC...
17:06:13 The VNC server is now running.
17:06:13

You chose to execute vnc with a password.


17:06:13 Please manually connect your vnc client to virtcook7.itso.ibm.com:1
(9.12.7.7) to begin the install.
17:06:13 Starting graphical installation.
```

11. Start a VNC client (for example, RealVNC) and connect to the server with your IP address with a **:1** appended to the end as shown in Figure 8-3.

12. When prompted for a password, enter the password that is specified in the `LNXADMIN PARM-RH6` file (**12345678** in the sample file). In the following example, Linux is being installed with the DNS name virtcook7.itso.ibm.com.

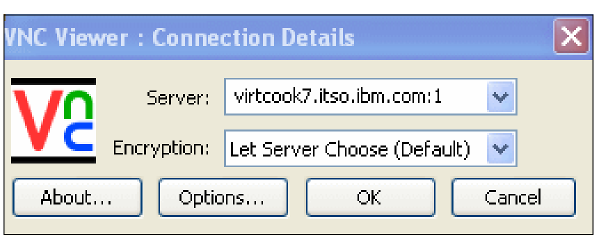


Figure 8-3 Connecting with VNC client

8.1.3 Stage 2 of the RHEL 6.4 installation

After you have connected using VNC, perform the following steps:

1. A splash screen appears as shown in Figure 8-4. Click `Next`.

Figure 8-4 RHEL 6.4 splash screen

2. You will be asked what type of devices to use, as shown Figure 8-5. For DASD devices, choose **Basic Storage Devices** and click **Next**.

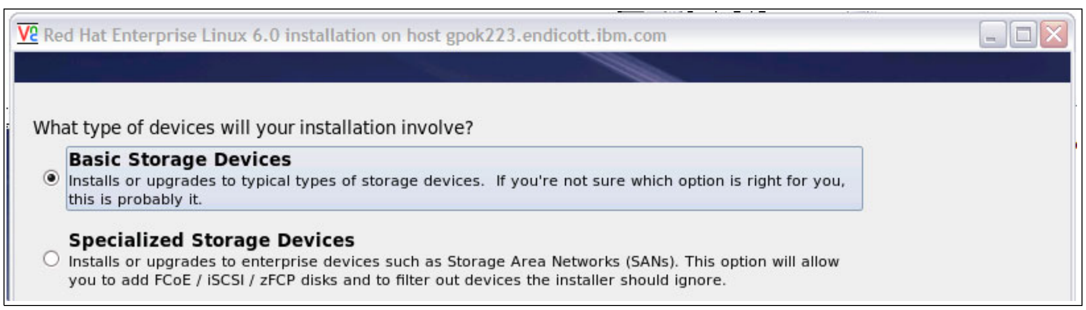


Figure 8-5 Device type panel

3. On the next window, the host name, read from the configuration file, should be correct. Click **Next**.
4. On the next window, select your time zone and click **Next**.
5. On the next window, set the root password and click **Next**. Do not forget it.
6. The installer now asks for the type of installation. Select the **Create Custom Layout** radio button as shown in Figure 8-6 on page 159. Click **Next**.

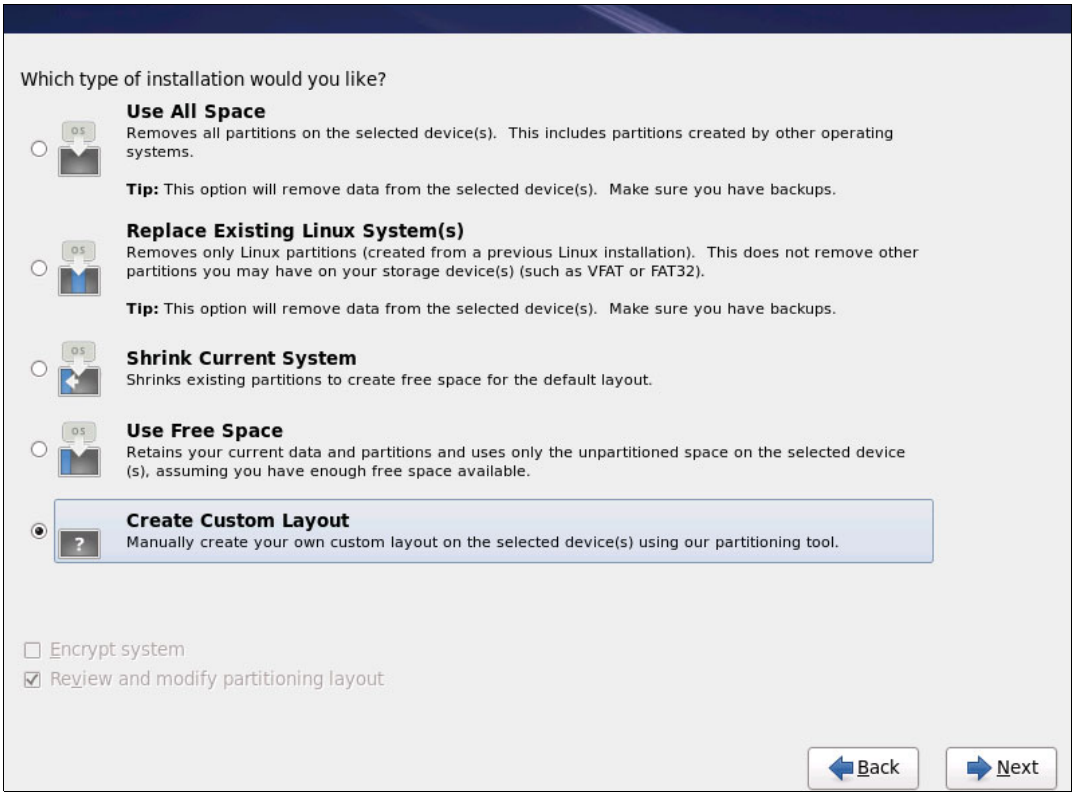


Figure 8-6 Creating custom disk layout

7. The *Please Select A Device* panel allows you to set up minidisks and virtual disks. Click the **Create** button, and a “*Create Storage”* window appears.
8. Accept the default of **Standard Partition** and click **Create**.

9. To create the root file system, select **dasdb**, and in the *Add a Partition* window, set it to fixed size and type 2500, as shown in Figure 8-7.

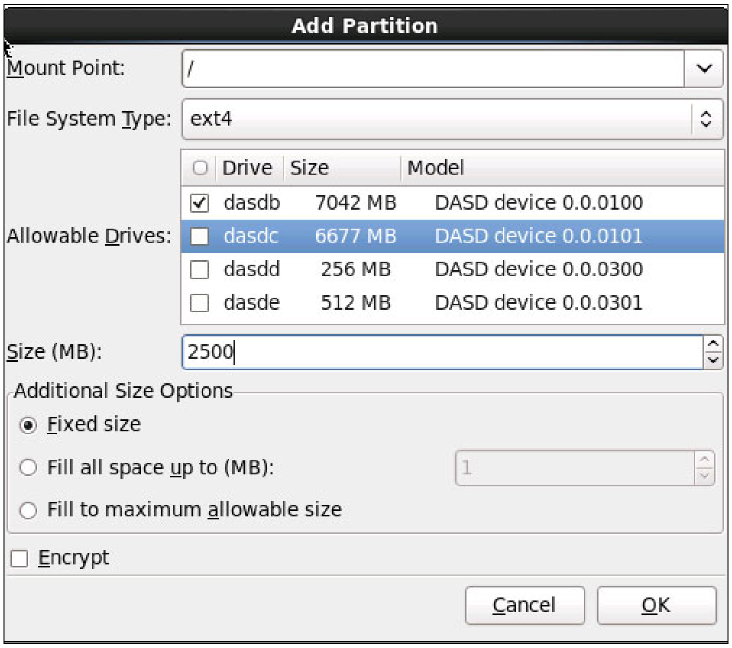


Figure 8-7 Adding a partition

10. Set up the remaining space on dasdb and dasdc (minidisks `100` and `101`) as LVM Physical Volumes. On the *Please Select A Device* panel, click the **Create** button.

11. Select the **LVM Physical Volume** radio button on the *Create Storage* panel and click **Create**.

12. On the *Add Partition* panel, clear all *Allowable Drives* except for *the volumes that are being set up*, as shown in Figure 8-8.

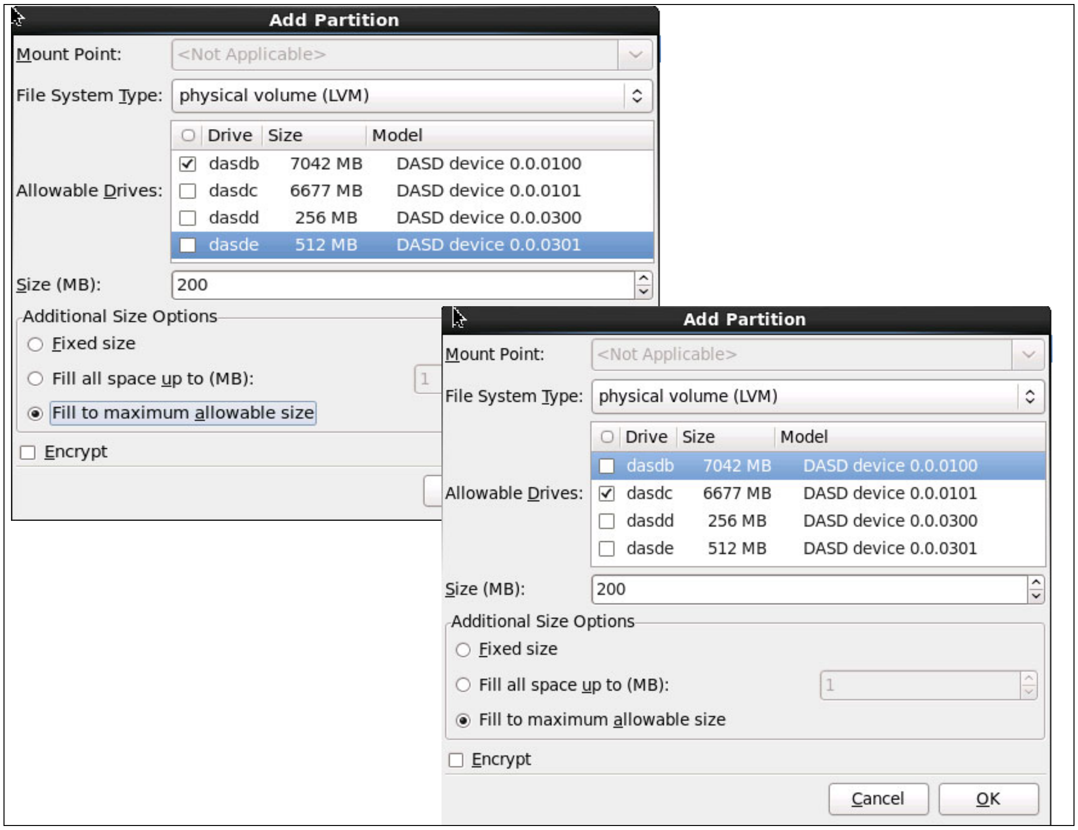


Figure 8-8 Creating physical volumes

- On the *Additional Size Options*, select the **Fill to maximum allowable size** radio button.
- Click **OK**.

13. The next step is to set up the LVM. Perform the following steps:

   a. Click **Create** again on the *Please Select A Device* panel and the *Create Storage* panel appears.

   b. On the *Create Storage* window, select the **LVM Volume Group** radio button and click **Create**. The *Make LVM Volume Group* panel appears, as shown on the left side of Figure 8-9.

   c. Set the *Volume Group Name* to **var_vg**, as shown at the center of Figure 8-9.

   d. Click **Add** under the *Logical Volumes* section. The *Make Logical Volume* panel appears.

   e. Set the *Mount Point* to **`/var`** and the *Logical Volume Name* to **var_lv**, as shown on the right side of Figure 8-9. Click **OK**.

   f. In the *Make LVM Volume Group* panel, click **OK**.

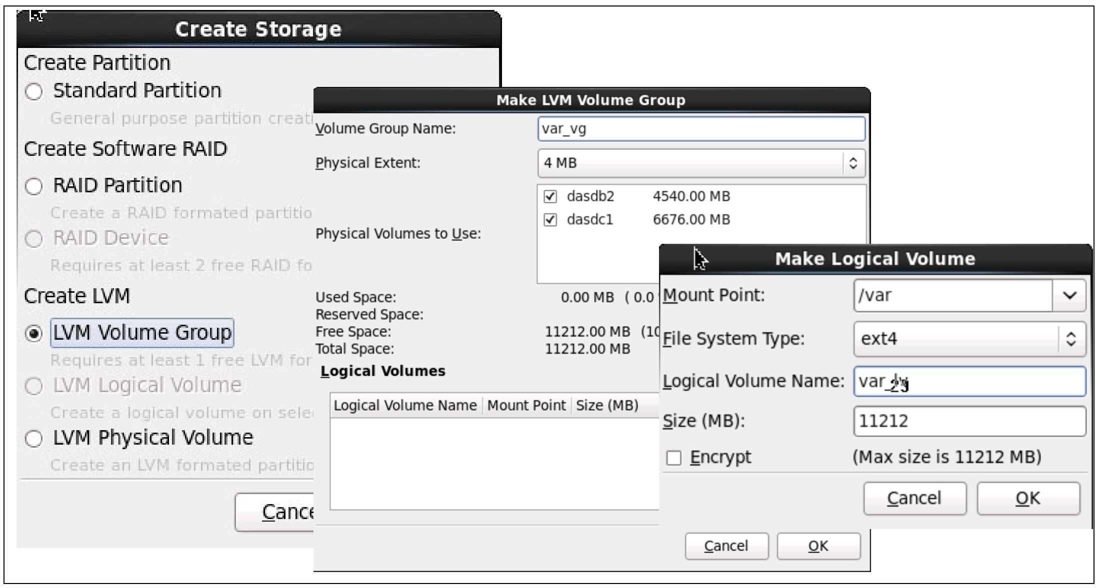
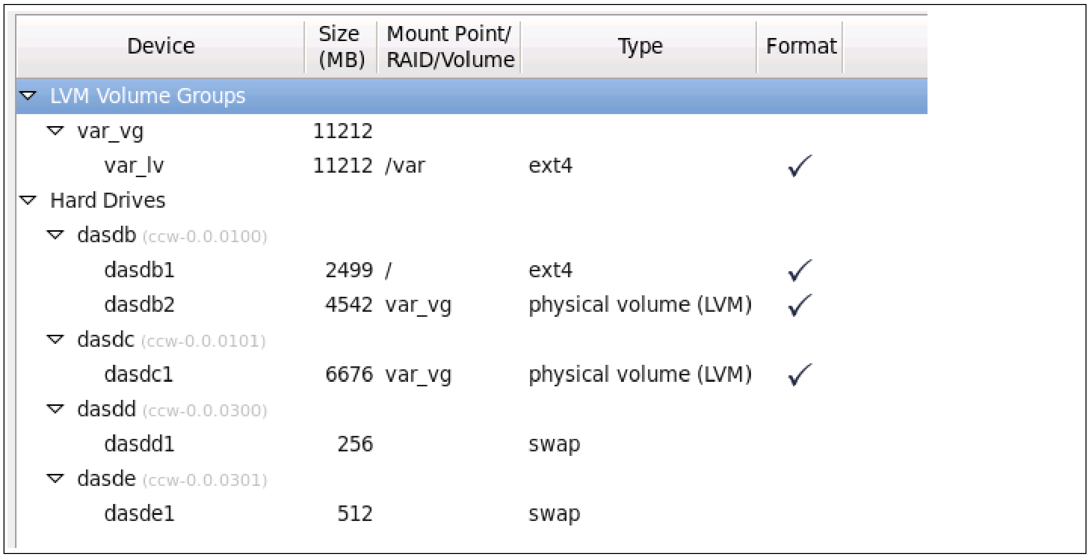


Figure 8-9 Creating a volume group and a logical volume

14. You will be returned to the *Please Select A Device* panel. It should look like what is shown in Figure 8-10 on page 163. Click **Next**.



Device	Size (MB)	Mount Point/ RAID/Volume	Type	Format
LVM Volume Groups				
var_vg	11212			
var_lv	11212	/var	ext4	✓
Hard Drives				
dasdb (ccw-0.0.0100)				
dasdb1	2499	/	ext4	✓
dasdb2	4542	var_vg	physical volume (LVM)	✓
dasdc (ccw-0.0.0101)				
dasdc1	6676	var_vg	physical volume (LVM)	✓
dasdd (ccw-0.0.0300)				
dasdd1	256		swap	
dasde (ccw-0.0.0301)				
dasde1	512		swap	

Figure 8-10 Disk summary

15. You might get a *Partition Warnings* panel. Click **Yes**.
16. On the *Writing storage confirmation to disk* panel, click **Write changes to disk**.

Important: If you see the window that is shown in Figure 8-11, you have to start the installation over. Be sure to use the **`dasdfmt`** command to format the minidisks, as described in 8.1.2, "Install RHEL 6.4" on page 154.

Figure 8-11 Symptom of known issue

17. You are asked for the type of software to be installed. Accept the default of **Basic Server** and click **Next**.
18. The installation process starts and a progress bar is shown. This should run for 5 - 10 minutes.
19. You are prompted to reboot. Click **Reboot.**

8.1.4 Boot your new Linux system from disk

A default system should now be installed onto minidisk `100` with a large logical volume mounted over `/var/`. Some points to note:

- Return to your z/VM 3270 session and your newly installed system should be re-IPLing automatically.
- Your system should continue to boot until a login prompt is presented.
- **Start an SSH session** as `root` to the new Linux administration system.
- Disconnect from the 3270 session:

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

The installation of RHEL 6.4 on the Linux administration system is now complete.

8.2 Configure the Linux administration system

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

1. "Copy RHEL 6.4 Installation tree to LNXADMIN"
2. "Configure yum" on page 165
3. "Turn off unneeded services" on page 166
4. "Configure the VNC server" on page 167
5. "Set system to halt on SIGNAL SHUTDOWN" on page 169
6. "Turn on the NFS server" on page 170
7. "Configure SSH keys" on page 170
8. "Insert the vmcp module and set system to log off" on page 171
9. "Reboot the system" on page 172
10. "Verify the changes" on page 172

8.2.1 Copy RHEL 6.4 Installation tree to LNXADMIN

Copy the RHEL 6.4 DVD ISO the Linux administration system, along with other files associated with this book. To do so, perform the following steps:

1. Create a local directory for the RHEL 6.4 installation tree:

```
# mkdir /var/nfs
```

2. Change into that directory:

```
# cd /var/nfs
```

3. Copy the RHEL installation tree from the PC NFS server (in this example the IP ***9.12.5.251***) with the **`scp -rp`** command:

```
# scp -rp 9.12.5.251:/srv/nfs/rhel64 .
The authenticity of host '9.12.5.251 (9.12.5.251)' can't be established.
RSA key fingerprint is 70:cf:ed:a8:98:4a:46:05:95:d9:72:b1:94:a9:16:25.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '9.12.5.251' (RSA) to the list of known hosts.
root@9.12.5.251's password:
...
```

This command takes some time, perhaps 5 - 15 minutes, depending on a number of factors.

The RHEL 6.4 installation tree should now be copied to the LNXADMIN system under /var/nfs/rhel64/.

8.2.2 Copy files associated with this book

To copy the files associated with this book to the Linux administration system, perform the following steps:

1. Change the directory to /var/nfs/ if you are not already there:

```
# cd /var/nfs
```

2. Recursively copy the files associated with this book from the PC NFS server with the **scp -rp** command. In this example, the IP address is ***9.12.5.251***:

```
# scp -rp 9.12.5.251:/var/nfs/SG248147 .
Password:
disclaimer.txt                                    100% 1461     1.4KB/s   00:00
README.txt                                        100% 1831     1.8KB/s   00:00
sample.conf-rh6                                   100%  235     0.2KB/s   00:00
profile.exec                                      100%  563     0.6KB/s   00:00
sample.parm-rh6                                   100%  168     0.2KB/s   00:00
swapgen.exec                                      100%   20KB  19.9KB/s   00:00
sample.parm-s11                                   100%  441     0.4KB/s   00:00
rhel64.exec                                       100%  504     0.5KB/s   00:00
sles11s3.exec                                     100%  380     0.4KB/s   00:00
cpformat.exec                                     100%   11KB  11.1KB/s   00:00
ssicmd.exec                                       100% 3449     3.4KB/s   00:00
callsm1.exec                                      100%   29KB  29.2KB/s   00:00
clone-1.0-11.s390x.rpm                            100%   12KB  11.8KB/s   00:00
clone.sh                                          100% 8848     8.6KB/s   00:00
linux5.xml                                        100%   11KB  11.2KB/s   00:00
jeos.tgz                                          100% 5106     5.0KB/s   00:00
boot.clone                                        100% 6262     6.1KB/s   00:00
```

The files associated with this book should now be copied to the Linux administration system under /var/nfs/SG248147/.

8.2.3 Configure yum

Configure **yum** so that it can install RPMs from the local installation tree. To do so, perform the following steps:

1. Create a file named rhel64.repo in the /etc/yum.repos.d directory:

```
# cd /etc/yum.repos.d
# vi rhel64.repo
[RHEL64]
name=Red Hat Enterprise Linux 6.4
baseurl=file:///var/nfs/rhel64/Server
```

2. Import the RPM key, which is included in the RHEL 6.4 DVD root directory:

```
# cd /var/nfs/rhel64
# rpm --import RPM-GPG-KEY-redhat-release
```

The RHEL 6.4 yum repository should now be configured.

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.

3. Verify that the new repository (often called *repo*) is accessible with the **yum grouplist** command:

```
# yum grouplist
Loaded plugins: product-id, security, subscription-manager
Updating certificate-based repositories.
SLoaded plugins: product-id, security, subscription-manager
This system is not registered to Red Hat Subscription Management. You can use
subscription-manager to register.
Setting up Group Process
Installed Groups:
   Additional Development
   Base
   Console internet tools
   Debugging Tools
   ...
   Zulu Support [zu]
Done
```

You are now ready to use **yum** to install or upgrade an RPM package. To install a package, use **yum -y install <packagename>**. Yum will conveniently install the packages specified and automatically resolve dependencies for you. You should not specify the package version on the command line, only the package name.

8.2.4 Turn off unneeded services

There are a number of services that are started in an RHEL 6.4 minimum system. In order to keep the Linux administration system as lean as possible, some of these can be turned off. To do so, perform the following steps:

1. Turn off each of the services using the following **chkconfig** commands:

```
# chkconfig iptables off
# chkconfig ip6tables off
# chkconfig auditd off
# chkconfig abrtd off
# chkconfig atd off
# chkconfig cups off
# chkconfig mdmonitor 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 about configuring **iptables** for NFS traffic, see the article at the following website:

http://www.redhat.com/magazine/010aug05/departments/tips_tricks

Or, access the Red Hat Knowledge Base:

https://access.redhat.com/site/articles/8756

Also, turning on a firewall is briefly discussed in section 11.1.3, "Turn on a firewall" on page 208.

2. You can choose to leave these services on, or turn off others. You can review which services are now configured to start in run level 3 with the following **chkconfig** command:

```
# chkconfig --list | grep 3:on
abrt-ccpp       0:off   1:off   2:off   3:on    4:off   5:on    6:off
autofs          0:off   1:off   2:off   3:on    4:on    5:on    6:off
blk-availability        0:off   1:on    2:on    3:on    4:on    5:on    6:off
certmonger      0:off   1:off   2:off   3:on    4:on    5:on    6:off
cpi             0:off   1:on    2:on    3:on    4:on    5:on    6:off
crond           0:off   1:off   2:on    3:on    4:on    5:on    6:off
dumpconf        0:on    1:on    2:on    3:on    4:on    5:on    6:on
haldaemon       0:off   1:off   2:off   3:on    4:on    5:on    6:off
kdump           0:off   1:off   2:off   3:on    4:on    5:on    6:off
lvm2-monitor    0:off   1:on    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
mon_statd       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
nfslock         0:off   1:off   2:off   3:on    4:on    5:on    6:off
portreserve     0:off   1:off   2:on    3:on    4:on    5:on    6:off
postfix         0:off   1:off   2:on    3:on    4:on    5:on    6:off
rhnsd           0:off   1:off   2:on    3:on    4:on    5:on    6:off
rhsmcertd       0:off   1:off   2:off   3:on    4:on    5:on    6:off
rpcbind         0:off   1:off   2:on    3:on    4:on    5:on    6:off
rpcgssd         0:off   1:off   2:off   3:on    4:on    5:on    6:off
rpcidmapd       0:off   1:off   2:off   3:on    4:on    5:on    6:off
rsyslog         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
sysstat         0:off   1:on    2:on    3:on    4:on    5:on    6:off
udev-post       0:off   1:on    2:on    3:on    4:on    5:on    6:off
```

8.2.5 Configure the VNC server

Often, applications require a graphical environment. The VNC server allows for a graphical environment to be set up easily by starting the **vncserver** service. To do so, perform the following steps:

1. Install the VNC server and associated packages with the following **yum** command:

```
# yum -y install tigervnc-server openmotif xterm xsetroot xorg-x11-xauth
...
Installed:
  openmotif.s390x 0:2.3.3-5.el6_3          tigervnc-server.s390x 0:1.1.0-5.el6
  xorg-x11-server-utils.s390x 0:7.5-13.el6  xorg-x11-xauth.s390x 1:1.0.2-7.1.el6
```

```
  xterm.s390x 0:253-1.el6

Dependency Installed:
  libXaw.s390x 0:1.0.11-2.el6              libXdmcp.s390x 0:1.1.1-3.el6
  libXmu.s390x 0:1.1.1-2.el6               libXp.s390x 0:1.0.0-15.1.el6
  libXpm.s390x 0:3.5.10-2.el6              libXxf86misc.s390x 0:1.0.3-4.el6
  libmcpp.s390x 0:2.7.2-4.1.el6            libxkbfile.s390x 0:1.0.6-1.1.el6
  mcpp.s390x 0:2.7.2-4.1.el6               xkeyboard-config.noarch 0:2.6-6.el6
  xorg-x11-fonts-misc.noarch 0:7.2-9.1.el6  xorg-x11-xkb-utils.s390x 0:7.7-4.el6

Complete!
```

2. The VNC server configuration file is `/etc/sysconfig/vncservers`. Edit the file by adding one line at the bottom:

```
# cd /etc/sysconfig
# vi vncservers
...
# VNCSERVERS="2:myusername"
# VNCSERVERARGS[2]="-geometry 800x600 -nolisten tcp -localhost"
VNCSERVERS="1:root"
```

3. Set a VNC password with the **vncpasswd** command. This password will be needed to connect to the VNC server:

```
# vncpasswd
Password: lnx4vm
Verify: lnx4vm
```

4. Stop the firewall:

```
# service iptables stop
iptables: Flushing firewall rules: [  OK  ]
iptables: Setting chains to policy ACCEPT: filter [  OK  ]
iptables: Unloading modules: [  OK  ]
```

5. Start the VNC server. This will create some initial configuration files under the `/root/.vnc/` directory:

```
# service vncserver start
Starting VNC server: 1:root xauth:  creating new authority file /root/.Xauthority

New 'virtcook7.itso.ibm.com:1 (root)' desktop is virtcook7.itso.ibm.com:1

Creating default startup script /root/.vnc/xstartup
Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/virtcook7.itso.ibm.com:1.log

[  OK  ]
```

6. The `/root/.vnc/` directory is where configuration files are kept. Change to that directory and list the files:

```
# cd /root/.vnc
# ls
passwd  virtcook7.itso.ibm.com:1.log  virtcook7.itso.ibm.com:1.pid  xstartup
```

7. The `xstartup` file is the script that is run when the VNC server starts and where the window manager is set. It is recommended that you change from the *Tab Window Manager*, **twm**, to the more usable Motif window manager, **mwm**:

```
# vi xstartup   // change last line
...
```

```
xsetroot -solid grey
vncconfig -iconic &
xterm -geometry 80x24+10+10 -ls -title "$VNCDESKTOP Desktop" &
mwm &
```

8. Restart the VNC server with the **service** command:

```
# service vncserver restart
Shutting down VNC server: 1:root [  OK  ]
Starting VNC server: 1:root
New 'virtcook7.itso.ibm.com:1 (root)' desktop is virtcook7.itso.ibm.com:1

Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/virtcook7.itso.ibm.com:1.log

[  OK  ]
```

9. You should now be able to use the VNC client to connect to the IP address of the Linux administration system with a **:1** appended. A sample session is shown in Figure 8-12.

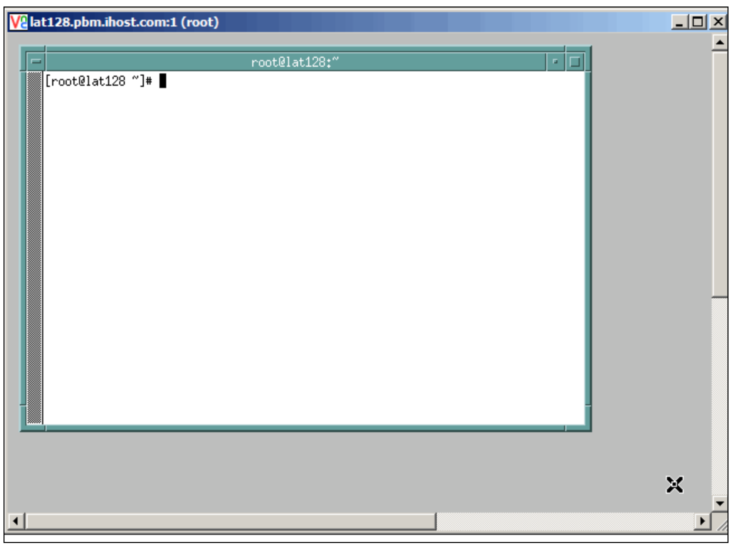


Figure 8-12 VNC client session to the VNC server

Note that the VNC server will not be started automatically across reboots. When you need a graphical environment, you can either start the **vncserver** service manually (recommended), or you can use **chkconfig** to enable automatic startup.

8.2.6 Set system to halt on SIGNAL SHUTDOWN

By default, RHEL 6.4 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. To set the system to halt, perform the following step:

1. Edit the `/etc/init/control-alt-delete.conf` file and change **shutdown -r** (reboot) to **shutdown -h** (halt):

```
# cd /etc/init
# vi control-alt-delete.conf
# control-alt-delete - emergency keypress handling
#
# This task is run whenever the Control-Alt-Delete key combination is
# pressed.  Usually used to shut down the machine.
```

```
start on control-alt-delete

exec /sbin/shutdown -h now "Control-Alt-Delete pressed"
```

After that change, when the system receives a **SIGNAL SHUTDOWN** from z/VM, the following message will be displayed:

```
Control-Alt-Delete pressed
```

8.2.7 Turn on the NFS server

The NFS server will be needed to export the RHEL 6.4 installation tree and the files associated with this book to the other virtual servers.

Enable NFS by using the following steps:

1. Edit the empty file `/etc/exports` and add the following two lines:

   ```
   # cd /etc
   # vi exports
   /var/nfs/rhel64            *(ro,sync)
   /var/nfs/SG248147          *(ro,sync)
   ```

   These two lines will cause NFS to export:

   – The `/var/nfs/rhel64/` directory, which contains the RHEL 6.4 installation files.
   – The `/var/nfs/SG248147/` directory, which has the files associated with this book.

2. Set the NFS server to start at boot time and for this session:

   ```
   # service nfs start
   Starting NFS services:  [  OK  ]
   Starting NFS quotas: [  OK  ]
   Starting NFS daemon: [  OK  ]
   Starting NFS mountd: [  OK  ]
   # chkconfig nfs on
   ```

3. Test mounting the two directories locally, then umount when successful:

   ```
   # mount localhost:/var/nfs/rhel64 /mnt
   # ls /mnt
   boot.cat                 RELEASE-NOTES-es-ES.html  RELEASE-NOTES-pt-BR.html
   EULA                     RELEASE-NOTES-fr-FR.html  RELEASE-NOTES-ru-RU.html
   ...
   # umount /mnt
   # mount localhost:/var/nfs/SG248147 /mnt
   # ls -F /mnt
   disclaimer.txt  README.txt  rhel64/  sles11sp2/  vm/
   # umount /mnt
   ```

In this section, you have turned on the NFS server and exported the RHEL 6.4 installation directory and the files associated with this book.

8.2.8 Configure SSH keys

SSH sessions are typically authenticated with passwords typed in from the keyboard. With SSH *key-based authentication*, sessions can be authenticated with public and private keys so that no password is needed. SSH key-based authentication can be set up from the Linux administration system (client) to the virtual servers. If the master image has a copy of Linux administration system's public key in the file `/root/.ssh/authorized_keys`, key-based authentication will work to the cloned virtual servers. To do this, create a new DSA key in the `/root/.ssh/` directory.

If it does not yet exist, first create it with the **mkdir** command:

```
# cd /root/.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:
54:23:16:2b:cf:bf:c7:97:b0:1a:2c:0f:b7:d2:5e:6d root@virtcook7.itso.ibm.com
The key's randomart image is:
+--[ DSA 1024]----+
|        +.o      |
|       . + .     |
|      . o        |
|       =         |
|        S        |
|         o  ..   |
|        o.=..oE. |
|        .=.=+.o  |
|         o*o .   |
+-----------------+
```

This creates a key pair where the file with the `.pub` suffix is the public key and the other file is the private key. The private key is only readable by root:

```
# ls -l id_dsa*
-rw-------. 1 root root 672 Jun  5 16:38 id_dsa
-rw-r--r--. 1 root root 617 Jun  5 16:38 id_dsa.pub
```

These files will be copied to the golden image later in the next chapter.

8.2.9 Insert the vmcp module and set system to log off

The **vmcp** module is used to issue CP commands. By default, it is not loaded at boot time. One way to accomplish this is to add the **modprobe vmcp** command to the file `/etc/rc.d/rc.local`, which is run at boot time.

When Linux is shut down, the default is for the virtual machine to remain logged on even though it is not running an operating system. It is more convenient for the user ID to be logged off, both at z/VM **SHUTDOWN** time and for getting a refreshed 3270 emulator session. The file `/etc/rc.d/rc.local` can again be used to make two calls to the **chshut** command.

To insert the **vmcp** module and to set the virtual machine to log off at shutdown time, edit the `/etc/rc.d/rc.local` file and add three lines at the end, as shown here:

```
# cd /etc/rc.d
# vi rc.local
#!/bin/sh
#
# This script will be executed *after* all the other init scripts.
# You can put your own initialization stuff in here if you don't
# want to do the full Sys V style init stuff.

touch /var/lock/subsys/local
modprobe vmcp
chshut halt vmcmd logoff
chshut poff vmcmd logoff
```

The **vmcp** command will now be available after the next reboot. The z/VM user ID should now be logged off when you halt or power off Linux.

8.2.10 Reboot the system

You should now reboot the system to test the changes:

```
# reboot
Broadcast message from root@virtcook7.itso.ibm.com
        (/dev/pts/0) at 7:27 ...

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

After your system comes back in a couple of minutes, start a new SSH session to the Linux administration system.

8.2.11 Verify the changes

You are now done customizing the Linux administration system. Perform the following steps to verify the configuration changes:

1. Start an SSH session to LNXADMIN.
2. Test the **vmcp** command with a CP command such as **QUERY NAMES**:

   ```
   # vmcp q n
   FTPSERVE - DSC , TCPIP    - DSC , DTCVSW2  - DSC , DTCVSW1  - DSC
   VMSERVP  - DSC , VMSERVR  - DSC , VMSERVU  - DSC , VMSERVS  - DSC
   OPERSYMP - DSC , DISKACNT - DSC , EREP     - DSC , LNXADMIN -L0006
   VSM     - TCPIP
   ```

3. Confirm that both VDisk swap spaces are operational:

   ```
   # swapon -s
   Filename                                Type            Size    Used    Priority
   /dev/dasdc1                             partition       262132  0       -1
   /dev/dasdb1                             partition       524276  0       -2
   ```

4. Verify that the NFS server is running:

   ```
   # service nfs status
   rpc.svcgssd is stopped
   rpc.mountd (pid 1368) is running...
   nfsd (pid 1431 1430 1429 1428 1427 1426 1425 1424) is running...
   rpc.rquotad (pid 1364) is running...
   ```

5. You can also choose to test shutting down LNXADMIN from a 3270 session to MAINT on SSI member 1. Use the **SIGNAL SHUTDOWN** command:

   ```
   ==> signal shutdown lnxadmin
   ```

   From the SSH session, you should see:

   ```
   Broadcast message from root@virtcook7.itso.ibm.com
           (unknown) at 16:57 ...

   The system is going down for halt NOW!
   Control-Alt-Delete pressed
   ```

6. Wait about a minute, and query the virtual machine. That it is logged off shows that the **chshut** commands are working:

```
Ready;
... after less than one minute ...
HCPSIG2113I User LNXADMIN has reported successful termination
==> q lnxadmin
08:09:48 HCPCQU045E LNXADMIN not logged on
Ready(00045); T=0.01/0.01 08:09:48
```

These steps show that the configuration changes made to the administrative Linux system have taken effect.

9

Installing and configuring the RHEL 6.4 golden image

"Reality is merely an illusion, albeit a very persistent one."

— Albert Einstein

In this chapter, you install the copy of RHEL 6.4 that will be cloned. This will be referred to as the *golden image*. It should be as lean as possible to be a generic virtual server and to fit comfortably on two 3390-3 DASDs, or about 4.4 GB.

In this section, you perform the following tasks:

- "Install the golden image" on page 175
- "Configure the golden image" on page 183

Chapter 4, "Configure an NFS/FTP server" on page 45, Chapter 5, "Install a z/VM SSI cluster" on page 57, Chapter 6, "Service z/VM" on page 103, and Chapter 8, "Install Red Hat Enterprise Linux on LNXADMIN" on page 151 must be completed before proceeding.

9.1 Install the golden image

In this section, you install an RHEL 6.4 golden image. It is installed onto the virtual machine `RH64GOLD`.

9.1.1 Create the RH64GOLD virtual machine

In this section, you define the `RH64GOLD` virtual machine to z/VM:

1. Log on to `MAINT`.
2. Edit the `USER DIRECT` file:

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

3. Go to the bottom of the file and add the definition for a new virtual machine named `RH64GOLD`. This virtual machine is given class G privilege only.

Be sure to replace the volume labels (***JM1264*** in this example) with the labels of your DASD:

```
USER RH64GOLD LNX4VM 256M 1G G
 INCLUDE LNXDFLT
 OPTION LNKNOPAS APPLMON
 MDISK 100 3390 0001 5008 JM1264 MR LNX4VM LNX4VM LNX4VM
 MDISK 101 3390 5009 5008 JM1264 MR LNX4VM LNX4VM LNX4VM
*
```

This Linux virtual machine will have the minidisks and virtual disks that are shown in Table 9-1.

Table 9-1 Minidisks to be defined

Minidisk	Description
100-101	Minidisks used to create the root file system, and logical volumes containing the other file systems
300-301	Virtual disk swap spaces that are not defined in USER DIRECT file, but by calls to the **SWAPGEN EXEC** in the user's **PROFILE EXEC**

4. Run the **DIRECTXA** command to bring the changes online:

```
==> directxa user
z/VM USER DIRECTORY CREATION PROGRAM - VERSION 6 RELEASE 3.0
EOJ DIRECTORY UPDATED AND ON LINE
HCPDIR494I User directory occupies 108 disk pages
Ready; T=0.01/0.02 11:07:39
```

You have now defined the virtual machine that will be the RHEL 6.4 golden Linux image.

9.1.2 Prepare RH64GOLD parameter files

Now that the RH64GOLD user is defined, you must create the PARM and CONF configuration files used by the RHEL 6.4 installer. To save time, copy the LNXADMIN PARM-RH6 and LNXADMIN CONF-RH6 files (described in Step 3 here), then make the necessary changes. Perform the following steps:

1. Logoff of MAINT if you are still logged on.
2. Log on to LNXMAINT.
3. The files LNXADMIN PARM-RH6 and LNXADMIN CONF-RH6 should exist on the 192 (D) disk as they were copied in 5.10.4, "Copy files associated with this book" on page 95. Copy the parameter and configuration files to new files with a file name of RH64GOLD. This can be done with the following **COPYFILE** command using wildcards:

```
==> copy lnxadmin * d RH64GOLD = =
```

4. Edit the new parameter file and change the CMSCONFFILE variable to point to the new configuration file:

```
==> x RH64GOLD parm-rh6
root=/dev/ram0 ro ip=off ramdisk_size=40000
CMSDASD=191 CMSCONFFILE=RH64GOLD.CONF-RH6
vnc vncpassword=12345678
```

5. Change the DASD, HOSTNAME, and IPADDR variables in the RH64GOLD CONF-RH6 configuration file. For these values, you might want to refer to the worksheet in section 2.9.7, "Host names and IP addresses worksheet" on page 33. Also, add one line with the METHOD= parameter pointing to the NFS server directory that you just set up on the Linux administration system. This precludes you from having to type in the NFS server

information in the **install** SSH session. Following is an example with the values used in this book:

```
==> x RH64GOLD conf-rh6
DASD=100-101,300-301
HOSTNAME=virtcook9.itso.ibm.com
NETTYPE=qeth
IPADDR=9.12.7.9
SUBCHANNELS=0.0.0600,0.0.0601,0.0.0602
NETMASK=255.255.240.0
SEARCHDNS=itso.ibm.com
METHOD=nfs:9.12.7.7:/var/nfs/rhel64
GATEWAY=9.12.4.1
DNS=9.12.6.7
MTU=1500
PORTNAME=DONTCARE
PORTNO=0
LAYER2=1
VSWITCH=1
```

You are now ready to start the golden image installation.

6. Logout LNXMAINT.

9.1.3 Install RHEL 6.4 on the golden image

Install Linux onto the RH64GOLD virtual machine by using the new installation tree exported from the Linux administration system.

Perform the following steps:

1. Log on to RH64GOLD. You should see authorization being granted to the virtual switch and virtual NIC at 600 being created and connected. The **PROFILE EXEC** from the LNXMAINT 192 disk should prompt you to IPL minidisk 100. Because there is nothing installed yet, answer **no**.

```
LOGON RH64GOLD
00: z/VM Version 6 Release 3.0, Service Level 0000 (64-bit),
00: built on IBM Virtualization Technology
00: There is no logmsg data
00: FILES:   NO RDR,   NO PRT,   NO PUN
00: LOGON AT 12:33:30 EDT THURSDAY 06/06/13
00: Command complete
00: NIC 0600 is created; devices 0600-0602 defined
00: NIC 0600 is connected to VSWITCH SYSTEM VSW1
z/VM V6.3.0    2013-06-04 12:50

DMSACP723I A (191) R/O
DMSACP723I C (592) R/O
DIAG swap disk defined at virtual address 300 (64988 4K pages of swap space)
DIAG swap disk defined at virtual address 301 (129980 4K pages of swap space)
Do you want to IPL Linux from minidisk 100? y/n
n
```

2. The default memory size of 256 MB is not enough to install RHEL 6.4. Set the memory size to 1 GB with the CP **DEFINE STORAGE** command:

```
==> def stor 1g
00: STORAGE = 1G
00: Storage cleared - system reset.
```

3. IPL CMS, and again answer **no**:

```
==> ipl cms
z/VM V6.3.0    2013-06-04 12:50

DMSACP723I A (191) R/O
DMSACP723I C (592) R/O
DIAG swap disk defined at virtual address 300 (64988 4K pages of swap space)
DIAG swap disk defined at virtual address 301 (129980 4K pages of swap space)
Do you want to IPL Linux from minidisk 100? y/n
n
```

4. To begin the installation program, run the **RHEL64 EXEC**:

```
==> rhel64
00:      NO FILES PURGED
00: RDR FILE 0001 SENT FROM RH64GOLD PUN WAS 0001 RECS 114K CPY  001 A NOHOLD NO
KEEP
00: RDR FILE 0005 SENT FROM RH64GOLD PUN WAS 0005 RECS 0003 CPY  001 A NOHOLD NO
KEEP
00: RDR FILE 0009 SENT FROM RH64GOLD PUN WAS 0009 RECS 229K CPY  001 A NOHOLD NO
KEEP
00: 0000003 FILES CHANGED
00: 0000003 FILES CHANGED
Initializing cgroup subsys cpuset
Initializing cgroup subsys cpu
Linux version 2.6.32-358.el6.s390x (mockbuild@s390-002.build.bos.redhat.com) (gc
c version 4.4.7 20120313 (Red Hat 4.4.7-3) (GCC) ) #1 SMP Tue Jan 29 12:06:31 ES
T 2013
setup: Linux is running as a z/VM guest operating system in 64-bit mode
```

5. There can be many, many screens of DASD I/O messages. Use the CP **TERM MORE** command to make the 3270 screens clear instantly:

```
==> #cp term more 0 0
```

6. The installation system should continue to boot and you should see the following messages:

```
Connect now to 9.12.7.9 and log in as user 'install' to start the installation.
E.g. using: ssh -x install@9.12.7.9
For VNC or text mode, disable X11 forwarding (recommended) with 'ssh -x'.
For X11, enable X11 forwarding with 'ssh -X'.

You may log in as the root user to start an interactive shell.
DEBUG kernel:eth0: no IPv6 routers present
```

Important! The message says to log in as the user **install**, however, there is an intermediate step. There is an issue where the Red Hat installer does not recognize disks that have been formatted with **CPFMTXA**.

7. If you followed all the steps in this book, this will be the case (if you previously used **dasdfmt** to format these minidisks, you can skip this step). You must first start an SSH session, login as root, and dasdfmt the disks. To do so, perform the following tasks:

   a. Start SSH session to the installation system and log in as **root**. A password will not be required.

   b. Invoke the **lsdasd** command and observe the disks:

```
# lsdasd
Bus-ID     Status      Name      Device  Type  BlkSz  Size      Blocks
==============================================================================
0.0.0100   n/f         dasdb     94:4    ECKD
```

```
0.0.0101   n/f        dasdc     94:8    ECKD
0.0.0300   active     dasdd     94:12   FBA   ???    256MB     ???
0.0.0301   active     dasde     94:16   FBA   ???    512MB     ???
```

Here, the minidisks `100 - 101` correspond to `dasdb` and `dasdc`.

c. Format the minidisks in parallel with the following **`for`** loop:

```
# for i in b c
> do
>   dasdfmt -b 4096 -y -f /dev/dasd$i &
> done
[1] 615
[2] 616
```

d. You might need to press **Enter** after a few minutes to see the jobs in the background complete:

```
Finished formatting the device.
Rereading the partition table... ok
Finished formatting the device.
Rereading the partition table... ok

[1]-  Done                    dasdfmt -b 4096 -y -f /dev/dasd$i
[2]+  Done                    dasdfmt -b 4096 -y -f /dev/dasd$i
```

8. Start an SSH session to the new in-memory Linux installer and log in as **install**.

```
login as: install
Welcome to the anaconda install environment 1.2 for zSeries
```

9. On the *Choose a Language* panel, set your language, tab to *Next*, and press **Enter**.
10. On the *Installation Method* screen, choose **NFS directory** for the installation method, and select **OK**.
11. The *NFS Setup* window should appear. Enter the IP address of the PC NFS server on the *NFS server name* line, and the path to the installation tree on the second line: `/var/nfs/rhel64/`.
12. Start a VNC client session to the name in the message and enter the password set in the configuration file (**12345678** in the examples):

```
16:56:23 Starting VNC...
16:56:25 The VNC server is now running.
16:56:25

You chose to execute vnc with a password.


16:56:25 Please manually connect your vnc client to virtcook9.itso.ibm.com:1 (9.12.7.9)
to begin the install.
16:56:25 Starting graphical installation.
```

13. A splash screen should appear. Click **`Next`**.
14. At the window asking for the type of devices, select **Basic Storage Devices** and click **Next**.
15. If you see A *Storage Device Warning* screen, click **Yes, Discard any data**. You will still be able to use the two virtual disk swap spaces at `300` and `301`.
16. At the window that sets the host name, the value read from the configuration file should be correct. Click **Next**.
17. Set the time zone and click **Next**.

18. Set the root password and click **Next**.
19. At the type of installation window, select **Create Custom Layout** and click **Next**. It is very important that you choose this option as described earlier.
20. On the resulting window, which shows all minidisks and virtual disks, click **Create**.
21. At the *Create Storage screen*, choose **Standard Partition**, and click **Create**.
22. At the *Add Partition screen,* as shown in Figure 9-1, set the *Mount Point* to the root file system (**/**), clear all drives except **dasdb**, and set the *Size (MB)* to **1024**. Click **OK**.

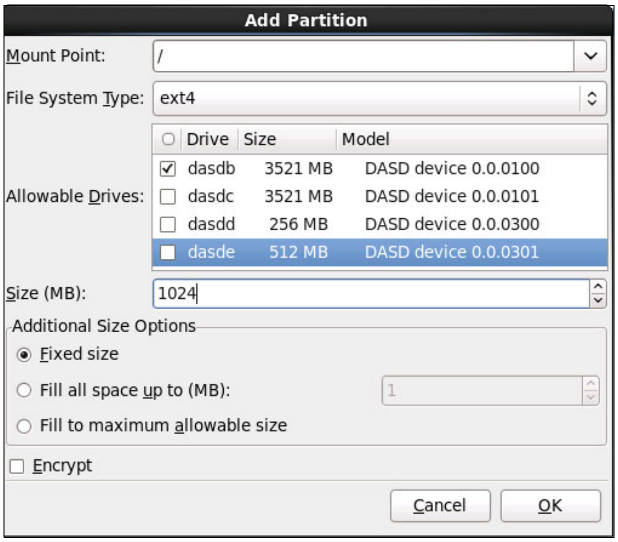


Figure 9-1 Defining the root file system

23. At the *Please Select a Device* screen, click **Create**.
24. At the *Create Storage* screen, select the **LVM Physical Volume** radio button and click **Create**.
25. On the Add Partition window, clear all *Allowable Drives* except **dasdb** (minidisk 100) and click the **Fill to maximum allowable size** radio button. Click **OK**.
26. Repeat the previous steps so that **dasdc** (minidisk 101) is also a physical volume.
27. The partitions **dasdd1** (virtual disk 300) and **dasde1** (virtual disk 301) should be recognized as swap spaces.
28. When you return to the *Please Select A Device* window, you should see the window that is shown in Figure 9-2 on page 181.

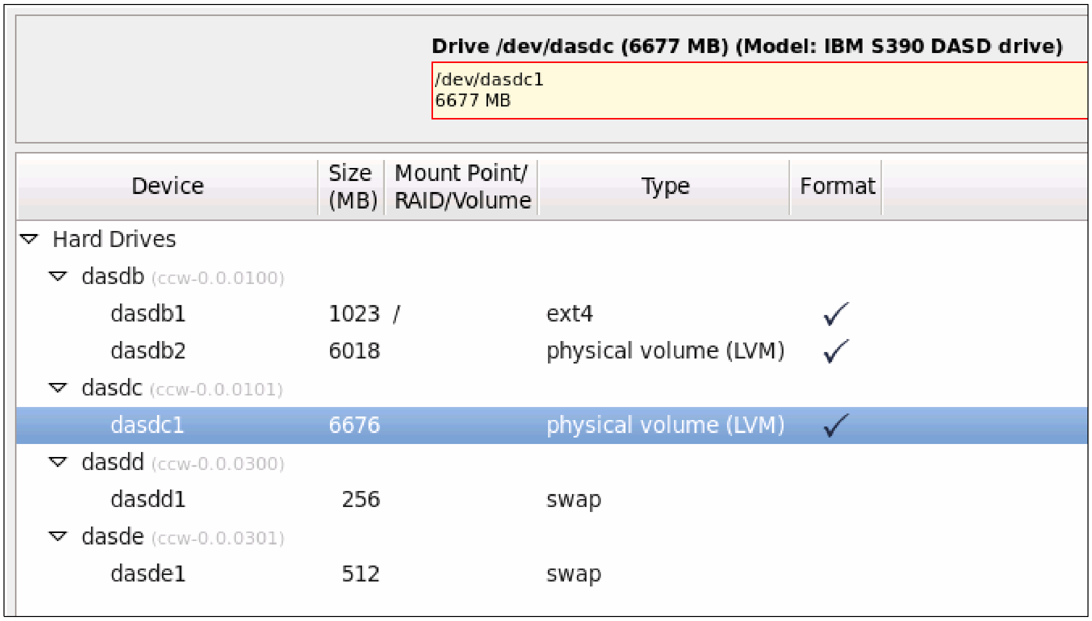


Figure 9-2 Initial disk allocation

29. At the *Please Select a Device* window, click **Create**.

30. On the *Create Storage* window, choose **LVM Volume Group** and click **Create**.

31. On the *Make LVM Volume Group* window, perform the following steps:

    a. Set the *Volume Group Name* to **system_vg**.
    b. Click **Add** in the *Logical Volumes* section.
    c. In the *Make Logical Volume* window, choose a mount point of `/tmp` and a size of 512 MB.

32. Repeat the above step for other file systems mounted at `/opt`, `/var`, and `/usr`. See Table 9-2 for the recommended logical volume layout and sizes.

Table 9-2 LVM logical volume layout

Mount point	**Logical volume name**	**Size (MB)**
`/tmp/`	tmp_lv	512
`/opt/`	opt_lv	512
`/var/`	var_lv	512
`/usr/`	usr_lv	2048

33. This results in about 720 MB of free space remaining in the volume group, as shown in Figure 9-3.

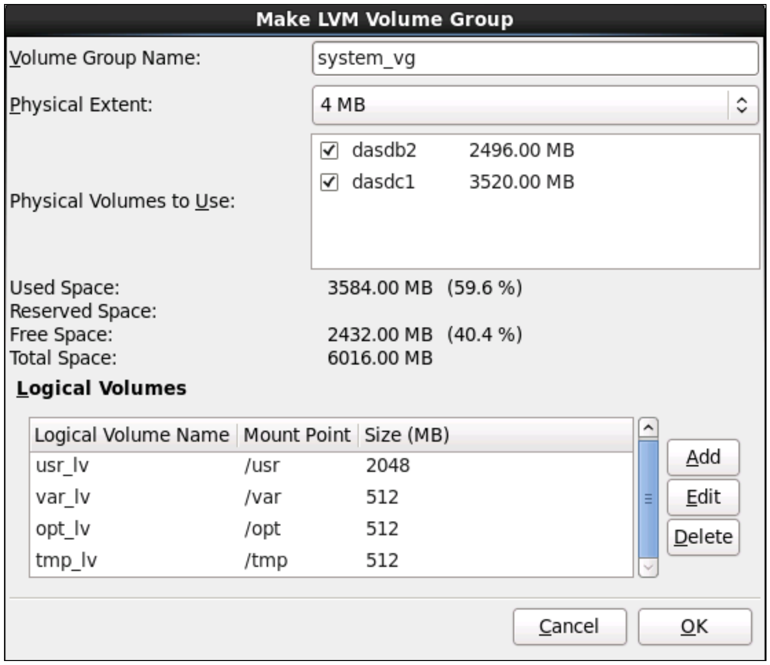
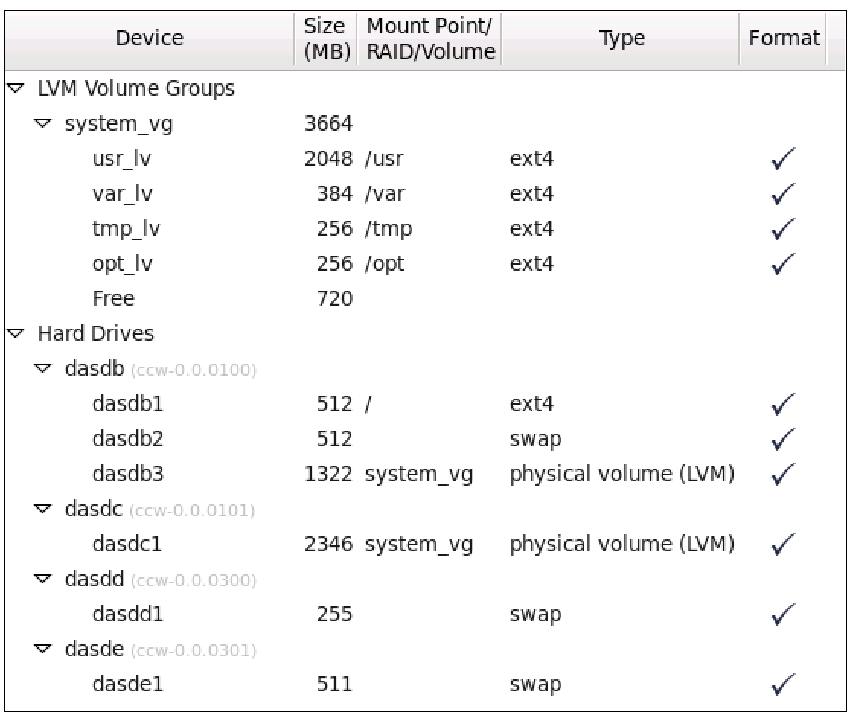


Figure 9-3 Defining a volume group and logical volumes

34. At the *Please Select A Device* window, click **Next**. You will see a *Format Warnings* window. Click **Format**. See Figure 9-4.



Device	Size (MB)	Mount Point/ RAID/Volume	Type	Format
LVM Volume Groups				
system_vg	3664			
usr_lv	2048	/usr	ext4	✓
var_lv	384	/var	ext4	✓
tmp_lv	256	/tmp	ext4	✓
opt_lv	256	/opt	ext4	✓
Free	720			
Hard Drives				
dasdb (ccw-0.0.0100)				
dasdb1	512	/	ext4	✓
dasdb2	512		swap	✓
dasdb3	1322	system_vg	physical volume (LVM)	✓
dasdc (ccw-0.0.0101)				
dasdc1	2346	system_vg	physical volume (LVM)	✓
dasdd (ccw-0.0.0300)				
dasdd1	255		swap	✓
dasde (ccw-0.0.0301)				
dasde1	511		swap	✓

Figure 9-4 Summary of file systems and swap spaces

35. On the *Format Warnings* panel, click **Format**.

36. At the *Writing storage configuration to disk* window, click **Write changes to disk**. The partitions, logical volumes, and file systems will be created on disk.
37. At the Software options section, accept the default of a **Basic Server** and click **Next**.
38. The installer takes about 5 - 10 minutes to install Linux. When complete, click **Reboot**. The system should be restarted from disk.
39. Start a new SSH session or restart the existing one to the RHEL 6.4 golden image. You might see a warning from PuTTY about a "POTENTIAL SECURITY BREACH". This is expected because a new set of SSH keys were generated for the same IP address. Click **Yes** to begin the session.

9.1.4 Verify the installation

Verify some settings with the following commands. You should see output similar to what is shown here:

```
# lsdasd
Bus-ID     Status      Name      Device  Type  BlkSz  Size      Blocks
==============================================================================
0.0.0100   active      dasda     94:0    ECKD  4096   3521MB    901440
0.0.0301   active      dasdb     94:4    FBA   512    512MB     1048576
0.0.0300   active      dasdc     94:8    FBA   512    256MB     524288
0.0.0101   active      dasdd     94:12   ECKD  4096   3521MB    901440

# swapon -s
Filename                                Type            Size    Used    Priority
/dev/dasdc1                             partition       262132  0       -1
/dev/dasdb1                             partition       524276  0       -2

# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/dasda1          1008M  182M  776M  19% /
tmpfs                 498M     0  498M   0% /dev/shm
/dev/mapper/system_vg-opt_lv
                      504M   17M  462M   4% /opt
/dev/mapper/system_vg-tmp_lv
                      504M   17M  462M   4% /tmp
/dev/mapper/system_vg-usr_lv
                      2.0G  1.3G  651M  67% /usr
/dev/mapper/system_vg-var_lv
                      504M   56M  423M  12% /var
```

This shows that the two minidisks and two virtual disks are enabled devices, that the two swap spaces are active.

9.2 Configure the golden image

Customize the golden image before cloning. The following high-level steps are recommended, though you can add or omit some steps:

1. "Configure automount of installation tree" on page 184
2. "Configure yum for online updates" on page 185
3. "Turn off unneeded services" on page 185

9.2.1 Configure automount of installation tree

Configure the Linux automount service 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.

To configure automount, perform the following steps:

1. Make a backup copy of the file /etc/auto.master:

```
# cd /etc
# cp auto.master auto.master.orig
```

2. Edit the file and add the following line at the bottom:

```
# vi auto.master   // add one line at the bottom
...
#
+auto.master
/var/nfs    /etc/auto.nfs
```

   The new line specifies that the file system mounted beneath the directory /var/nfs/ will be configured in the file /etc/auto.nfs.

3. Create that auto.nfs file and add one line that points to the RHEL 6.4 installation tree that is NFS-exported:

```
# vi auto.nfs
rhel64     -ro,hard,intr 9.12.7.7:/var/nfs/rhel64
```

   This line specifies that beneath /var/nfs/ (in auto.master), when the directory rhel64/ (field 1) is accessed, the *automounter* will use the specified options (field 2) to mount the directory (field 3).

4. Create the /nfs/ directory. Restart the **autofs** service to pick up the new configuration. Then, list the contents of the /var/nfs/rhel64/ directory. Even though this directory does not exist as a local file system, it is automatically mounted when referenced:

```
# mkdir /var/nfs
# service autofs reload
Reloading maps
```

5. Invoke the following two commands:

```
# ls /var/nfs
# ls /var/nfs/rhel64
boot.cat                 RELEASE-NOTES-en-US.html  RELEASE-NOTES-pt-BR.html
...
```

   Do you notice anything unusual in these two commands? This is an attribute of automount: The first command suggests that there are neither files or directories under /var/nfs/: however, the second command lists the contents of /var/nfs/rhel64/. This behavior is expected because the directory that is *auto-mounted* is not accessed with NFS until it is referenced.

The /var/nfs/rhel64/ directory should now be automatically mounted when referenced.

9.2.2 Configure yum for online updates

Configure **yum** so it can install RPMs from the *auto-mounted* installation tree. The configuration is identical to the Linux administration system because in both instances the installation tree is in the /var/nfs/rhel64/ directory. However, on the Linux administration system this directory is local, while on the golden image (and later the clones), the directory is *auto-mounted*. To configure **yum**, perform the following steps:

1. You could create a file named rhel64.repo in the /etc/yum.repos.d directory again, or you could copy the same file from the Linux administration system that you created previously. Use the **scp** command to copy the existing file:

```
# cd /etc/yum.repos.d
# scp virtcook7:/etc/yum.repos.d/rhel64.repo .
The authenticity of host 'virtcook7,9.12.7.7' can't be established.
RSA key fingerprint is a0:a8:85:f6:01:62:11:de:77:1d:8b:5e:b9:c6:5f:b1.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'virtcook7,9.12.7.7' (RSA) to the list of known hosts.
root@virtcook7's password:
rhel64.repo                                   100%   81     0.1KB/s   00:00
```

2. Type the file to verify the contents with the **cat** command:

```
# cat rhel64.repo
[RHEL64]
name=Red Hat Enterprise Linux 6.4
baseurl=file:///var/nfs/rhel64/Server
```

3. Import the RPM GPG key so that **yum** knows you are installing official Red Hat packages. The Red Hat GPG key is in the installation tree. Import the key with the following command:

```
# rpm --import /var/nfs/rhel64/RPM-GPG-KEY-redhat-release
```

The last command again shows that automount is working. The **yum** tool should now be configured. It will be tested in the next section.

9.2.3 Turn off unneeded services

As with the golden image, follow the steps in 8.2.4, “Turn off unneeded services” on page 166. Following is a summary of the commands you must execute to do this:

```
# chkconfig iptables off
# chkconfig ip6tables off
# chkconfig auditd off
# chkconfig abrtd off
# chkconfig atd off
# chkconfig cups off
# chkconfig mdmonitor off
```

Verify that these services are turned off with the **chkconfig --list** command:

```
# chkconfig --list | grep 3:on
abrt-ccpp         0:off   1:off   2:off   3:on    4:off   5:on    6:off
autofs            0:off   1:off   2:off   3:on    4:on    5:on    6:off
blk-availability  0:off   1:on    2:on    3:on    4:on    5:on    6:off
certmonger        0:off   1:off   2:off   3:on    4:on    5:on    6:off
cpi               0:off   1:on    2:on    3:on    4:on    5:on    6:off
crond             0:off   1:off   2:on    3:on    4:on    5:on    6:off
dumpconf          0:on    1:on    2:on    3:on    4:on    5:on    6:on
```

```
haldaemon        0:off   1:off   2:off   3:on    4:on    5:on    6:off
kdump            0:off   1:off   2:off   3:on    4:on    5:on    6:off
lvm2-monitor     0:off   1:on    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
mon_statd        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
nfslock          0:off   1:off   2:off   3:on    4:on    5:on    6:off
portreserve      0:off   1:off   2:on    3:on    4:on    5:on    6:off
postfix          0:off   1:off   2:on    3:on    4:on    5:on    6:off
rhnsd            0:off   1:off   2:on    3:on    4:on    5:on    6:off
rhsmcertd        0:off   1:off   2:off   3:on    4:on    5:on    6:off
rpcbind          0:off   1:off   2:on    3:on    4:on    5:on    6:off
rpcgssd          0:off   1:off   2:off   3:on    4:on    5:on    6:off
rpcidmapd        0:off   1:off   2:off   3:on    4:on    5:on    6:off
rsyslog          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
sysstat          0:off   1:on    2:on    3:on    4:on    5:on    6:off
udev-post        0:off   1:on    2:on    3:on    4:on    5:on    6:off
```

9.2.4 Configure the VNC server

It is now necessary to configure the VNC server. To do this, perform the following steps:

1. Install the VNC server and associated packages with the following **yum** command:

```
# yum -y install tigervnc-server openmotif xterm xsetroot xorg-x11-xauth
...
Installed:
  openmotif.s390x 0:2.3.3-5.el6_3          tigervnc-server.s390x 0:1.1.0-5.el6
  xorg-x11-server-utils.s390x 0:7.5-13.el6  xorg-x11-xauth.s390x 1:1.0.2-7.1.el6
  xterm.s390x 0:253-1.el6

Dependency Installed:
  libXaw.s390x 0:1.0.11-2.el6              libXdmcp.s390x 0:1.1.1-3.el6
  libXmu.s390x 0:1.1.1-2.el6               libXp.s390x 0:1.0.0-15.1.el6
  libXpm.s390x 0:3.5.10-2.el6              libXxf86misc.s390x 0:1.0.3-4.el6
  libmcpp.s390x 0:2.7.2-4.1.el6            libxkbfile.s390x 0:1.0.6-1.1.el6
  mcpp.s390x 0:2.7.2-4.1.el6               xkeyboard-config.noarch 0:2.6-6.el6
  xorg-x11-fonts-misc.noarch 0:7.2-9.1.el6  xorg-x11-xkb-utils.s390x 0:7.7-4.el6

Complete!
```

2. The VNC server configuration file is `/etc/sysconfig/vncservers`. Edit the file by adding one line at the bottom:

```
# cd /etc/sysconfig
# vi vncservers
...
# VNCSERVERS="2:myusername"
# VNCSERVERARGS[2]="-geometry 800x600 -nolisten tcp -localhost"
VNCSERVERS="1:root"
```

3. Set a VNC password with the **vncpasswd** command. This password will be needed to connect to the VNC server:

```
# vncpasswd
Password: lnx4vm
Verify: lnx4vm
```

4. Stop the firewall:

```
# service iptables stop
iptables: Flushing firewall rules: [  OK  ]
iptables: Setting chains to policy ACCEPT: filter [  OK  ]
iptables: Unloading modules: [  OK  ]
```

5. Start the VNC server. This creates some initial configuration files under the `/root/.vnc/` directory:

```
# service vncserver start
Starting VNC server: 1:root xauth:  creating new authority file /root/.Xauthority

New 'virtcook9.itso.ibm.com:1 (root)' desktop is virtcook9.itso.ibm.com:1

Creating default startup script /root/.vnc/xstartup
Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/virtcook9.itso.ibm.com:1.log

[  OK  ]
```

6. The `/root/.vnc/` directory is where configuration files are kept. Change to that directory and list the files:

```
# cd /root/.vnc
# ls
passwd  virtcook9.itso.ibm.com:1.log  virtcook9.itso.ibm.com:1.pid  xstartup
```

7. The `xstartup` file is the script that is run when the VNC server starts and where the window manager is set. It is recommended that you change from the Tab Window Manager, **twm**, to the more usable Motif window manager, **mwm**:

```
# vi xstartup   // change last line
...
xsetroot -solid grey
vncconfig -iconic &
xterm -geometry 80x24+10+10 -ls -title "$VNCDESKTOP Desktop" &
mwm &
```

8. Restart the VNC server with the **service** command:

```
# service vncserver restart
Shutting down VNC server: 1:root [  OK  ]
Starting VNC server: 1:root
New 'virtcook9.itso.ibm.com:1 (root)' desktop is virtcook9.itso.ibm.com:1

Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/virtcook9.itso.ibm.com:1.log

[  OK  ]
```

9. You should now be able to use the VNC client to connect to the IP address of the Linux administration system with a **:1** appended. A sample session is shown in Figure 9-5.

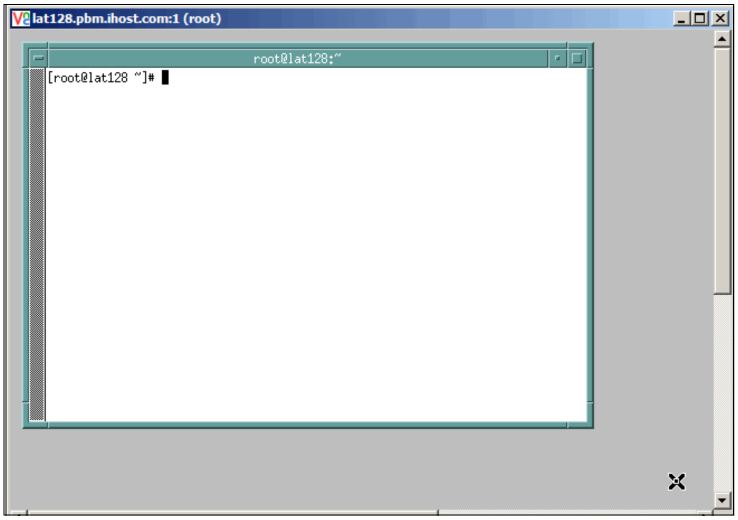


Figure 9-5 VNC client session to the VNC server

The VNC server will not be started automatically across reboots. When you need a graphical environment, you can either start the **vncserver** service manually (recommended), or you can use **chkconfig** to enable automatic startup.

9.2.5 Set system to halt on SIGNAL SHUTDOWN

RHEL 6.4 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 (shut down). To do this, edit the `/etc/init/control-alt-delete.conf` file and change **shutdown -r** (reboot) to **shutdown -h** (halt), as shown here:

```
# cd /etc/init
# vi control-alt-delete.conf
# control-alt-delete - emergency keypress handling
#
# This task is run whenever the Control-Alt-Delete key combination is
# pressed.  Usually used to shut down the machine.

start on control-alt-delete

exec /sbin/shutdown -h now "Control-Alt-Delete pressed"
```

This change will be picked up when the system is rebooted.

9.2.6 Configure settings at boot time

When Linux is shut down, the default is for the virtual machine to remain logged on even though it is not running an operating system. It is more convenient for the virtual machine to be logged off, both at z/VM **SHUTDOWN** time and for getting a refreshed 3270 emulator session. This can be modified with the **chshut** command and the parameters **halt** and **poff** (power off).

In order for z/VM 6.3 to relocate guests between single system image (SSI) members, there must not be any links to CMS disks. The `LNXDFLT PROFILE` creates links to minidisks and assigns virtual device numbers `190`, `191`, `19D`, and `19E`. These addresses must be detached. The **vmcp** module is used to accomplish this.

To do these two tasks, edit the `/etc/rc.d/rc.local` file, which is run at boot time:

```
# cd /etc/rc.d
# vi rc.local
```

And add the following lines:

```
#!/bin/sh
#
# This script will be executed *after* all the other init scripts.
# You can put your own initialization stuff in here if you don't
# want to do the full Sys V style init stuff.

touch /var/lock/subsys/local
chshut halt vmcmd logoff
chshut poff vmcmd logoff
modprobe vmcp
vmcp det 190
vmcp det 191
vmcp det 19d
vmcp det 19e
```

The z/VM virtual machine should now be logged off when you halt or power off Linux.

9.2.7 Configure SSH keys

Recall that you generated SSH public and private keys on the Linux administration system in section 8.2.8, “Configure SSH keys” on page 170. Now it is time to copy these keys from the Linux administration system to the golden image:

1. Change into the `/root/.ssh/` directory:

   ```
   # cd /root/.ssh
   ```

   If it does not exist, create it with the **mkdir** command.

2. Copy the Linux administration system’s public key to `authorized_keys` using **scp**. In this example the IP address of the Linux administration system is ***9.12.7.7***:

   ```
   # scp 9.12.7.7:/root/.ssh/id_dsa.pub authorized_keys
   root@9.12.7.7's password:
   id_dsa.pub                                  100%  617     0.6KB/s   00:00
   ```

   This will allow the Linux administration system to initiate an encrypted SSH connection to the Linux server using *key-based authentication* (so there will be no need to type the root password).

3. Switch to, or start and SSH session on the Linux administration system. Use the **ssh** command to the golden image to issue the **hostname** command. You should not be prompted for a password:

   ```
   [root@virtcook7 ~]# ssh virtcook9 hostname
   The authenticity of host 'virtcook9 (9.12.7.9)' can't be established.
   RSA key fingerprint is 1c:39:ce:14:b8:b3:06:bb:45:ea:bd:29:5c:09:1f:0a.
   Are you sure you want to continue connecting (yes/no)? yes
   Warning: Permanently added 'virtcook9,9.12.7.9' (RSA) to the list of known hosts.
   virtcook9.itso.ibm.com
   ```

You should not have to type the RHEL 6.4 golden image password.

You should now have key-based authentication from the Linux administration system to the golden image configured.

9.2.8 Disable cgroup memory

Linux Control Groups, or *cgroups*, are collections of processes with the same criteria. They allow you to limit, account, and isolate resource usage.

You can set `cgroup_disable=memory` as a boot option to disable memory resource control, which can result in up to 1% more free memory. Unless you know why you need this feature, the following steps are recommended to disable it:

1. Make a backup copy of the `/etc/zipl.conf` file:

```
# cd /etc
# cp zipl.conf zipl.conf.orig
```

2. Add the kernel parameter `cgroup_disable=memory`:

```
# vi zipl.conf
[defaultboot]
timeout=5
default=linux-2.6.32-358.el6.s390x
target=/boot/
[linux-2.6.32-358.el6.s390x]
        image=/boot/vmlinuz-2.6.32-358.el6.s390x
        ramdisk=/boot/initramfs-2.6.32-358.el6.s390x.img
        parameters="root=/dev/disk/by-path/ccw-0.0.0100-part1 cgroup_disable=memory
rd_NO_LUKS rd_DASD=0.0.0301 rd_DASD=0.0.0300 rd_NO_MD  KEYTABLE=us
SYSFONT=latarcyrheb-sun16 crashkernel=auto rd_DASD=0.0.0100 rd_NO_LVM
rd_NO_DM LANG=en_US.UTF-8"
```

Note: The `parameters=` line is shown on four lines for clarity. It is just one line.

3. Run the **zipl** command for the change to take effect:

```
# zipl
Using config file '/etc/zipl.conf'
Building bootmap in '/boot/'
Building menu 'rh-automatic-menu'
Adding #1: IPL section 'linux-2.6.32-358.el6.s390x' (default)
Preparing boot device: dasda (0100).
Done.
```

Linux cgroup memory should now be disabled.

9.2.9 Other configuration changes

You can consider other configuration changes. Of course, you can take an iterative approach: Start with this set of changes, clone some Linux images and test, then bring the golden image back up, make more changes, and reclone.

Whether you are on the first pass of configuration or not, refer to the following sections to consider other changes for performance and availability-related issues:

- 13.1, “Register your system with RHN” on page 235
- 26.2, “Set up Memory Hotplugging” on page 488
- 26.4, “Hardware cryptographic support for OpenSSH” on page 493

9.2.10 Reboot the system

Now **reboot** to test your changes:

```
# reboot
Broadcast message from root@virtcook9.itso.ibm.com
        (/dev/pts/0) at 15:26 ...

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

9.2.11 Verify the changes

You are now done customizing the RHEL 6.4 golden image. When the system comes back, perform the following steps to verify the changes that you made:

1. Start an SSH session to the golden image as **root** and check a few settings.
2. Use the **df** command to display your file systems:

```
# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/dasda1          1008M  183M  775M  20% /
tmpfs                 498M     0  498M   0% /dev/shm
/dev/mapper/system_vg-opt_lv
                      504M   17M  462M   4% /opt
/dev/mapper/system_vg-tmp_lv
                      504M   17M  462M   4% /tmp
/dev/mapper/system_vg-usr_lv
                      2.0G  1.3G  623M  68% /usr
/dev/mapper/system_vg-var_lv
                      504M   86M  393M  18% /var
```

3. Confirm that your swap spaces are operational and that the two virtual disks have priority over the one minidisk:

```
# swapon -s
Filename                                Type            Size    Used    Priority
/dev/dasdc1                             partition       262132  0       -1
/dev/dasdb1                             partition       524276  0       -2
```

4. Verify the shutdown settings with the **lsshut** command:

```
# lsshut
Trigger          Action
========================
Halt             vmcmd ("logoff")
Power off        vmcmd ("logoff")
Reboot           reipl
Restart          stop
Panic            stop
```

You can choose to confirm other settings.

Congratulations. You have now successfully installed and configured the golden image. This image will normally be shut down or *quiesced*. You are now ready to clone the golden image to a new virtual server.

10

Configure RHEL 6.4 for cloning

"*It has become appallingly obvious that our technology has exceeded our humanity.*"

— Albert Einstein

By now you should have installed and configured `LNXADMIN`, the Linux administration system, and `RH64GOLD`, the *golden image*. The Linux administration system must be up and running. In this chapter, the following tasks are described:

- "Define three new virtual machines" on page 194
- "Clone a virtual server manually" on page 195
- "Clone a virtual server automatically" on page 200

A virtual server is cloned manually so you will better understand the steps, and it is cloned automatically to speed up the process.

A block diagram of this process is displayed in Figure 10-1 on page 194.

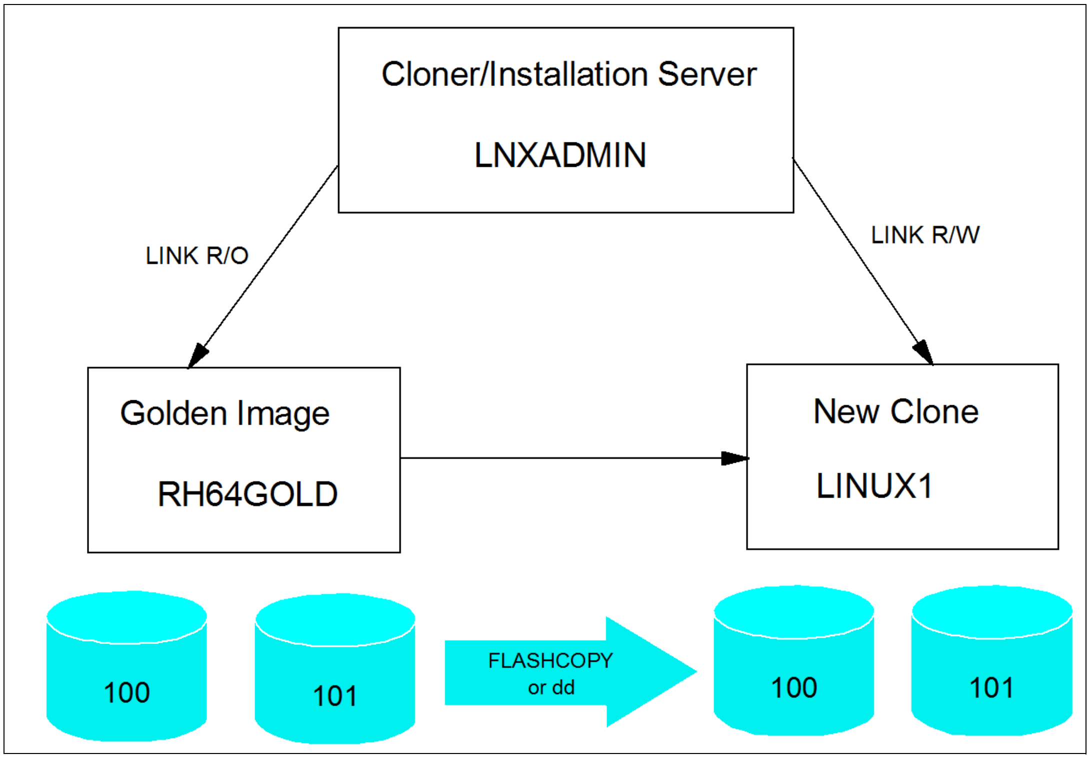


Figure 10-1 Cloning architecture

The top of the figure shows the Linux administration system that is running on the LNXADMIN virtual machine. In order to **FLASHCOPY** or **dd**, the LNXADMIN virtual machine requires a **LINK** to the source minidisks that RH64GOLD owns and the destination minidisks that LINUX1 owns. The figure shows that the **LINK** statement is issued as read-only (**RR**) for the source and read/write (**W**) for the target. The virtual disk-based swap spaces at addresses 300 and 301 are defined in-memory, therefore, they do not need to be copied.

10.1 Define three new virtual machines

In this section, you define new virtual machines that will be targets to clone to. In this example, they are named LINUX1, LINUX2, and LINUX3 (each user ID suffix corresponds to the last dotted decimal octet in the IP address associated).

To do so, perform the following steps:

1. Log on to MAINT and edit the USER DIRECT file to add new virtual machines for Linux:

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

2. Go to the bottom of the file and add the following lines. In this example, the user IDs will be LINUX1, LINUX2, and LINUX3 with a password of LNX4VM. They will default to have 512 MB of memory but can be set up to 1 GB. They will have only G privilege class (General user). They will each have two 5008 cylinder minidisks. In this example, the 3390-9 assigned have labels of ***JM1266***, ***JM1267***, and ***JM1268***. These disks were formatted in section 5.8.4, “Format DASD for minidisks” on page 84:

   ```
   *
   USER LINUX1 LNX4VM 512M 1G G
    INCLUDE LNXDFLT
    MDISK 100 3390 0001 5008 JM1266 MR LNX4VM LNX4VM LNX4VM
   ```

```
 MDISK 101 3390 5009 5008 JM1266 MR LNX4VM LNX4VM LNX4VM
*
USER LINUX2 LNX4VM 512M 1G G
 INCLUDE LNXDFLT
 MDISK 100 3390 0001 5008 JM1267 MR LNX4VM LNX4VM LNX4VM
 MDISK 101 3390 5009 5008 JM1267 MR LNX4VM LNX4VM LNX4VM
*
USER LINUX3 LNX4VM 512M 1G G
 INCLUDE LNXDFLT
 MDISK 0100 3390 0001 5008 JM1268 MR READPASS WRITPASS MULTPASS
 MDISK 0101 3390 5009 5008 JM1268 MR READPASS WRITPASS MULTPASS
```

3. You might need to add the new volumes to the $ALLOC$ virtual machine so cylinder 0 will not show up in the disk map as a gap.
4. Again check for gaps and overlaps:

```
==> diskmap user (doends
==> x user diskmap
====> all /gap/|/overlap/
...
```

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

```
==> directxa user
z/VM USER DIRECTORY CREATION PROGRAM - VERSION 6 RELEASE 3.0
EOJ DIRECTORY UPDATED AND ON LINE
HCPDIR494I User directory occupies 109 disk pages
Ready; T=0.01/0.02 07:58:39
```

The new virtual machines have now been created.

10.2 Clone a virtual server manually

Before using the **clone** script 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 virtual machine, RH64GOLD in this example, has a root file system located on minidisks 100-101.
- The target virtual machine, LINUX1, has identically sized minidisks 100-101.
- The **vmcp** command is available to issue z/VM CP commands.
- The z/VM **FLASHCOPY** command can be used but the Linux **dasdfmt** and **dd** commands will also work.

Given these assumptions, one set of steps that can be used to clone Linux is as follows:

1. “Link the source and target disks” on page 195.
2. “Copy the source to the target disks” on page 196
3. “Modify the new root file system” on page 197
4. “IPL the target system” on page 199

10.2.1 Link the source and target disks

To link the source and target disks, perform the following steps:

1. Shut down the RHEL 6.4 golden image if it is still running:

```
# shutdown -h now
Broadcast message from root@virtcook9.itso.ibm.com
        (/dev/pts/0) at 8:04 ...

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

   The virtual machine should automatically be logged off.

2. Start an SSH session to the Linux administration system, `virtcook7` (`LNXADMIN`) in this example, as `root`.
3. Link the source disks, `RH64GOLD 100` and `101`, read-only as virtual devices `1100` and `1101` with the **`LINK`** command:

```
# vmcp link rh64gold 100 1100 rr
# vmcp link rh64gold 101 1101 rr
```

4. Link the target disks, `LINUX1 100` and `101`, multi-read (read/write if no other virtual machine has write access) as virtual devices `2100` and `2101`:

```
# vmcp link linux1 100 2100 mr
# vmcp link linux1 101 2101 mr
```

10.2.2 Copy the source to the target disks

To copy the source to the target disks, perform the following steps:

1. **If you have** the **`FLASHCOPY`** feature available, use it with the **`vmcp`** command:

```
# vmcp flashcopy 1100 0 end to 2100 0 end
Command complete: FLASHCOPY 1100 0 5007 TO 2100 0 5007
# vmcp flashcopy 1101 0 end to 2101 0 end
Command complete: FLASHCOPY 1101 0 5007 TO 2101 0 5007
```

2. **If you do not have** the **`FLASHCOPY`** feature, enable the `1100-1101` and `2100-2101` disks with the **`chccwdev -e`** command:

```
# chccwdev -e 1100-1101,2100-2101
Setting device 0.0.1100 online
Done
Setting device 0.0.1101 online
Done
Setting device 0.0.2100 online
Done
Setting device 0.0.2101 online
Done
```

   a. Check if the dasds are online and check the device labels and determine the newly created device nodes with the **`lsdasd`** command:

```
# lsdasd
Bus-ID     Status      Name      Device  Type  BlkSz  Size      Blocks
============================================================================
==
0.0.0100   active      dasda     94:0    ECKD  4096   7042MB    1802880
0.0.0301   active      dasdb     94:4    FBA   512    512MB     1048576
0.0.0300   active      dasdc     94:8    FBA   512    256MB     524288
0.0.0101   active      dasdd     94:12   ECKD  4096   6676MB    1709280
0.0.2100   n/f         dasde     94:16   ECKD
0.0.1100   active      dasdf     94:20   ECKD  4096   3521MB    901440
0.0.1101   active      dasdg     94:24   ECKD  4096   3521MB    901440
```

```
0.0.2101   n/f         dasdh     94:28   ECKD
```

b. In this example, the source minidisks (1100-1101) are named /dev/dasdf and /dev/dasdg, while the target minidisks (2100-2101) are named /dev/dasdh and /dev/dasde. Format the target devices with the **dasdfmt** command using a 4096 byte (4 KB) block size:

```
# dasdfmt -b 4096 -y -f /dev/dasde
Finished formatting the device.
Rereading the partition table... ok
# dasdfmt -b 4096 -y -f /dev/dasdh
Finished formatting the device.
Rereading the partition table... ok
```

c. Now that the devices have been formatted, you can copy the volumes of the golden image with the **dd** command, again using a block size of 4 K (4096) bytes:

```
# dd if=/dev/dasdf of=/dev/dasde bs=4096
901440+0 records in
901440+0 records out
3692298240 bytes (3.7 GB) copied, 40.9585 s, 90.1 MB/s

# dd if=/dev/dasdg of=/dev/dasdh bs=4096
901440+0 records in
901440+0 records out
3692298240 bytes (3.7 GB) copied, 38.6801 s, 95.5 MB/s
```

d. Bring the devices offline so the new file systems will be recognized when brought back online:

```
# chccwdev -d 1100-1101,2100-2101
Setting device 0.0.1100 offline
Done
Setting device 0.0.1101 offline
Done
Setting device 0.0.2100 offline
Done
Setting device 0.0.2101 offline
Done
```

e. Detach the source disks because they are no longer needed:

```
# vmcp det 1100-1101
1100-1101 DETACHED
```

10.2.3 Modify the new root file system

To modify the newly copied root file system, perform the following steps:

1. The target root file system is on the LINUX1 100 disk, which is linked as virtual address 2100. Activate it with the **chccwdev -e** command:

```
# chccwdev -e 2100
Setting device 0.0.2100 online
Done
```

2. Use the **lsdasd** command to show the minidisks that are accessible. The target root file system is on the disk accessed as virtual device address 2100:

```
# lsdasd
Bus-ID     Status      Name      Device  Type  BlkSz  Size      Blocks
```

```
================================================================================
0.0.0100   active      dasda     94:0    ECKD  4096   7042MB    1802880
0.0.0301   active      dasdb     94:4    FBA   512    512MB     1048576
0.0.0300   active      dasdc     94:8    FBA   512    256MB     524288
0.0.0101   active      dasdd     94:12   ECKD  4096   6676MB    1709280
0.0.2100   active      dasde     94:16   ECKD  4096   3521MB    901440
```

3. In this example, the target disk device is `/dev/dasde` and the target disk first partition is `/dev/dasde1`. Mount the target root file system over the directory `/mnt/`:

```
# mount /dev/dasde1 /mnt
```

4. Observe that this appears to be a root file system:

```
# ls /mnt
bin   cgroup  etc   lib    lost+found  misc  net  proc  sbin     srv  tmp  var
boot  dev     home  lib64  media       mnt   opt  root  selinux  sys  usr
```

5. The networking values that must be changed are the IP address and host name in the files `/etc/sysconfig/network` and `/etc/sysconfig/network-scripts/ifcfg-eth0`.

6. Observe the contents of these files on the newly copied disks:

```
# cat /mnt/etc/sysconfig/network
NETWORKING=yes
HOSTNAME=virtcook9.itso.ibm.com
GATEWAY=9.12.4.1

# cat /mnt/etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE="eth0"
BOOTPROTO="static"
DNS1="9.12.6.7"
DOMAIN="itso.ibm.com"
GATEWAY="9.12.4.1"
IPADDR="9.12.7.9"
MTU="1500"
NETMASK="255.255.240.0"
NETTYPE="qeth"
NM_CONTROLLED="yes"
ONBOOT="yes"
OPTIONS="layer2=1 portno=0"
PORTNAME="DONTCARE"
SUBCHANNELS="0.0.0600,0.0.0601,0.0.0602"
TYPE="Ethernet"
UUID="ca14a846-7001-4586-b113-198ad74f7121"
```

7. Change the host name on the target disks in the file `/etc/sysconfig/network`:

```
# cd /mnt/etc/sysconfig
# vi network
NETWORKING=yes
HOSTNAME=virtcook1.itso.ibm.com
GATEWAY=9.12.4.1
```

8. Change the IP address in the file `/etc/sysconfig/network-scripts/ifcfg-eth0`:

```
# cd network-scripts
# vi ifcfg-eth0
DEVICE="eth0"
BOOTPROTO="static"
DNS1="9.12.6.7"
DOMAIN="itso.ibm.com"
GATEWAY="9.12.4.1"
IPADDR="9.12.7.1"
MTU="1500"
```

```
NETMASK="255.255.240.0"
NETTYPE="qeth"
NM_CONTROLLED="yes"
ONBOOT="yes"
OPTIONS="layer2=1 portno=0"
PORTNAME="DONTCARE"
SUBCHANNELS="0.0.0600,0.0.0601,0.0.0602"
TYPE="Ethernet"
```

9. Now that the target disks have been copied and modified, they can be detached. Change to the default directory with the **cd** command, use the **sync** command to flush the disks and the **umount** command to unmount the modified root file system:

```
# cd
# sync
# umount /mnt
```

10. Set the LINUX1 100 disks offline with the **chccwdev** command:

```
# chccwdev -d 2100
Setting device 0.0.2100 offline
Done
```

11. Detach the two target disks using the CP **DETACH** command:

```
# vmcp det 2100-2101
2100-2101 DETACHED
```

You should now be ready to IPL the manually cloned system.

10.2.4 IPL the target system

The modified target system should now be ready to IPL. To do so, perform the following steps:

1. Log on to LINUX1.

```
00: z/VM Version 6 Release 3.0, Service Level 0000 (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 10:08:43 EDT FRIDAY 06/07/13
00: Command complete
00: NIC 0600 is created; devices 0600-0602 defined
00: NIC 0600 is connected to VSWITCH SYSTEM VSW1
z/VM V6.3.0    2013-06-04 12:50
```

2. Press **Enter** to IPL CMS. The **PROFILE EXEC** will ask you if you want to IPL from minidisk 100. Type **y** for yes and Linux should boot:

```
DMSACP723I A (191) R/O
DMSACP723I C (592) R/O
DIAG swap disk defined at virtual address 300 (64988 4K pages of swap space)
DIAG swap disk defined at virtual address 301 (129980 4K pages of swap space)
Do you want to IPL Linux from minidisk 100? y/n
y
00: zIPL v1.8.2-49.el6 interactive boot menu
00:
00:  0. default (linux-2.6.32-358.el6.s390x)
00:
00:  1. linux-2.6.32-358.el6.s390x
00:
00: Note: VM users please use '#cp vi vmsg <input>'
00:
00: Please choose (default will boot in 5 seconds):
```

```
00: Booting default (linux-2.6.32-358.el6.s390x)...
Initializing cgroup subsys cpuset
Initializing cgroup subsys cpu
Linux version 2.6.32-358.el6.s390x (mockbuild@s390-002.build.bos.redhat.com) (gc
c version 4.4.7 20120313 (Red Hat 4.4.7-3) (GCC) ) #1 SMP Tue Jan 29 12:06:31 ES
T 2013
...
Red Hat Enterprise Linux Server release 6.4 (Santiago)
Kernel 2.6.32-358.el6.s390x on an s390x

virtcook1 login:
```

Your new system should come up cleanly using the modified IP address and host name. If it does, then congratulations. You have now cloned a Linux system manually. You can look around the new system. It should be identical to the golden image except for the IP address and host name.

Next, you learn how to do it automatically. You use the `LINUX1` virtual machine again. To clone, the target virtual machine must be logged off. You could shut down the new system cleanly, but because you will be cloning again, it does not matter. Go to the 3270 session and log off the `LINUX1` virtual machine:

```
==> #cp log
```

10.3 Clone a virtual server automatically

Now that you have cloned a server manually and better understand the steps, you can use the **`clone`** script to clone automatically. To do so, perform the following steps:

1. "Install the clone RPM" on page 200
2. "Create a configuration file for cloning" on page 200
3. "Use the clone script" on page 202

10.3.1 Install the clone RPM

Perform the following steps on the `LNXADMIN` installation server:

1. Open an SSH session as root to the system running on `LNXADMIN`.
2. Install the clone script RPM:

```
# rpm -ivh /var/nfs/SG248147SG248147/rhel64/clone-1.0-11.s390x.rpm
Preparing...                ########################################### [100%]
   1:clone                  ########################################### [100%]
```

10.3.2 Create a configuration file for cloning

For each Linux guest you want to clone, you must create a configuration file that you can use to customize the image after cloning. To do that for `LINUX1`, perform the following steps:

1. Review the configuration file `/etc/sysconfig/clone`, which is used to set variables:

```
# cat /etc/sysconfig/clone
# AUTOLOG - If set to "y" the script will autolog the cloned
#           image after the cloning is completed.  If it is
#           set to "n" the image will not autolog the cloned
#           image.
AUTOLOG=y

# PROMPT  - This will set if the script should prompt the user for
```

```
#           confirmation before cloning.  If set to "y" the user
#           will be prompted to continue.  If set to "n" the script
#           will run without confirmation.
PROMPT=y

# CLONE_MNT_PT - This specifies the location on the filesystem
#                that the cloned root filesystem should be mounted
#                to.  If the directory does not exist it will be
#                created the first run.
CLONE_MNT_PT=/mnt/clone

# CLONE_METHOD - This is used to determine what method you want to use
#                for cloning.  It can have a value of AUTO, which will first
#                attempt FLASHCOPY then fall back to dd, or DD which will
#                only try to perform a Linux dd command.
CLONE_METHOD=auto

# BLACKLIST    - List of z/VM user IDs forbidden to be used as clone targets.
#                It's a good idea to add your master server here, so it doesn't
#                become a clone target by mistake.
#                Format: BLACKLIST="userA userB userC ..."
BLACKLIST=""
```

In the following example, this configuration file is not modified because all defaults are acceptable.

2. Copy and then edit the supplied sample configuration file to reflect the values of the new Linux system:

   ```
   # cd /etc/clone
   # cp rhel.conf.sample linux2.conf
   ```

3. Edit the new configuration file with the appropriate values for your system. If the new Linux image is going to be on the same network as the golden image, you are likely to have to change only two variables: The IP address (`IPADDR`) and the DNS host name (`HOSTNAME`). In the following example, the IP address is set to ***9.12.7.2*** and the host name to ***virtcook2.itso.ibm.com***:

   ```
   # vi linux2.conf
   # Define the DASD that should be included as a part
   # of the clone.
   DASD=100,101
   DASD_ROOT=100
   VG_NAME=
   LV_ROOT=
   AUTOLOG=y

   # Define networking information that will be used for the host.
   IPADDR=9.12.7.2
   SUBCHANNELS=0.0.0600,0.0.0601,0.0.0602
   HOSTNAME=virtcook2.itso.ibm.com
   NETTYPE=qeth
   NETMASK=255.255.240.0
   NETWORK=9.12.4.0
   SEARCHDNS=itso.ibm.com
   BROADCAST=9.12.7.255
   GATEWAY=9.12.4.1
   DNS=9.12.6.7
   MTU=1500
   ```

   Notes on the previous values:

– The DASD= line sets the range of minidisks that will be copied. You can enter dashes (**-**) to specify address ranges, or commas (**,**) to specify specific disks. Make the range following DASD= is one continuous block of text with no spaces added.
– The DASD_ROOT= line sets the minidisk that contains the root file system.
– The VG_NAME= line sets the volume group name if the root file system of the source system is on a logical volume.
– The LV_ROOT= line sets the logical volume name of the root file system.
– The AUTOLOG=y line specified that the target virtual machine will be started after cloning.

You are now ready to clone to this new virtual machine.

10.3.3 Use the clone script

To use the clone script, perform the following steps:

1. Go back to an SSH session to the Linux administration system.
2. Verify that the **clone** script is in your PATH with the **which** command:

```
# which clone
/usr/sbin/clone
```

3. The **clone** script can operate in two modes. The first where the DASD information is provided on the command line, and the second where the DASD information is included in the new virtual machine's configuration file. Running clone with no arguments prints the following help information:

```
# clone
Usage: clone [-v] sourceID targetID [rootMinidisk [minidisk1 minidisk2..]]
    Switches
       -v Verbose output
    Required
       sourceID the z/VM user id you want to clone from
       targetID  the z/VM user id you want to clone to
    Optional
       rootMinidisk the minidisk address that contains the root filesystem
       minidisk1..n additional minidisks that should be copied
```

The sourceID is the z/VM virtual machine of the RHEL 6.4 golden image (RH64GOLD in this example) and targetID is the z/VM ID of the target (LINUX2 in this example). These values are *always* required.

In this example, DASD is set to 100-101, which implies that minidisks located at virtual addresses 100 and 101 are copied. The 300 and 301 virtual disks are omitted because **SWAPGEN** automatically creates them each time the user logs on. The DASD_ROOT value specifies which one of these minidisks contains the Linux root file system (/).

The script exits if either the golden image or the clone image is logged in. The script first attempts to copy the disks with **FLASHCOPY** via the **vmcp** module or command. If an error is returned, the script falls back to using Linux **dasdfmt** and **dd** commands. Finally, the script boots the new Linux image via the **xautolog** command. It takes less than a minute to clone with **FLASHCOPY** support and 5 - 15 minutes with **dd**.

4. Run the **clone** script with the verbose switch (**-v**) to add output:

```
# clone -v rh64gold linux2
Invoking CP command: QUERY rh64gold
Invoking CP command: QUERY linux1

This will copy disks from rh64gold to linux1
```

```
Host name will be:  virtcook2.itso.ibm.com
IP address will be: 9.12.7.2
Do you want to continue? (y/n):
y
```

The script ensures that the golden image (source) virtual machine and the target virtual machine exist and are logged off. Then, it confirms the order of the cloning and displays information collected from the `/etc/clone/linux2.conf` file. Following this, it asks if you are sure that you want to overwrite the disks on the target virtual machine.

Next, the script links to the golden image minidisk and the target minidisk. The golden image minidisks are linked at virtual address `FFFE`, and the target minidisks are linked as `FFFF`. The `FFFE` links are read-only and the `FFFF` links are read/write. With the links in place, the script issues a **`FLASHCOPY`** command to copy the source `100` and `101` minidisks to the target `100` and `101` minidisks. The script then detaches the links. If **`FLASHCOPY`** fails, the script falls back to the Linux **`dasdfmt`** and **`dd`** commands:

```
Cloning rh64gold to linux2 ...
Copying minidisks...
Invoking CP command: QUERY VIRTUAL fffe
Invoking CP command: LINK rh64gold 100 fffe RR
Invoking CP command: QUERY VIRTUAL ffff
Invoking CP command: LINK linux2 100 ffff W
Invoking CP command: FLASHCOPY fffe 0 END ffff 0 END
100 disk copied ...
Invoking CP command: DETACH fffe
Invoking CP command: DETACH ffff
Invoking CP command: QUERY VIRTUAL fffe
Invoking CP command: LINK rh64gold 101 fffe RR
Invoking CP command: QUERY VIRTUAL ffff
Invoking CP command: LINK linux2 101 ffff W
Invoking CP command: FLASHCOPY fffe 0 END ffff 0 END
101 disk copied ...
Invoking CP command: DETACH fffe
Invoking CP command: DETACH ffff
Updating cloned image ...
Invoking CP command: QUERY VIRTUAL ffff
Invoking CP command: LINK linux2 100 ffff W
Modifying networking info under /mnt/clone...
Regenerating SSH keys in /mnt/clone/etc/ssh/ ...
Invoking CP command: DETACH ffff
Invoking CP command: XAUTOLOG linux2
Booting linux2
Successfully cloned rh64gold to linux2
```

The SSH keys are regenerated in such a way that they are unique for the new virtual server. The new root file system is then unmounted, set offline, and detached: In the final section, the `LINUX2` virtual machine is logged on via **`XAUTOLOG`**. Because the shared `PROFILE EXEC` detects that the virtual machine is in a disconnected mode, it carries out an IPL of Linux from minidisk `100`.

5. Start an SSH session to the newly cloned Linux server.

> **Note:** If the clone script fails, check the following settings:
>
> - The configuration contains all of the correct information in `/etc/clone/`
> - No other users have links to the clone's read/write disks

11

Create RHEL 6.4 appliances

"*The secret to creativity is knowing how to hide your sources.*"

— Albert Einstein

After you have the ability to clone generic Linux servers, it is relatively easy to customize them for specific tasks given the variety of open source tools included with Linux distributions. These customized virtual servers can be thought of as *appliances*.

This chapter describes four types of appliances:

- "Create a web server appliance" on page 206
- "Create an application development appliance" on page 210
- "Create an LDAP server appliance" on page 214
- "Create a file server appliance" on page 225

The sections that follow do not go into the theory or detail on the five 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 Create a web server appliance

This section describes how to install and configure a virtual web server using the following tasks:

- "Install Apache RPMs"
- "Test Apache" on page 207
- "Turn on a firewall" on page 208
- "Configure SSL" on page 209
- "Populate the website" on page 210

11.1.1 Install Apache RPMs

To install the Apache Web Server, perform the following steps:

1. Start an SSH session as root to the first target server that you cloned. In this example, it is ***virtcook1*** running on `LINUX1`.
2. Verify that the Apache RPMs are not installed:

```
# rpm -qa | grep httpd
```

   No output shows that no RPMs with the string `httpd` are installed.

3. **SSH** into the IP address of the new `LINUX1` server. Install the following Apache RPMs with the **`yum -y install`** command. The **`-y`** flag prevents the "Is this OK" question:

```
# yum -y install httpd httpd-manual
...
Installed:
  httpd.s390x 0:2.2.15-26.el6        httpd-manual.noarch 0:2.2.15-26.el6

Dependency Installed:
  apr.s390x 0:1.3.9-5.el6_2               apr-util.s390x 0:1.3.9-3.el6_0.1
  apr-util-ldap.s390x 0:1.3.9-3.el6_0.1   httpd-tools.s390x 0:2.2.15-26.el6

Complete!
```

4. Verify that the RPMs were installed:

```
# rpm -qa | grep httpd
httpd-manual-2.2.15-26.el6.noarch
httpd-2.2.15-26.el6.s390x
httpd-tools-2.2.15-26.el6.s390x
```

5. 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
```

6. Start the Apache Web Server:

```
# service httpd start
Starting httpd: [  OK  ]
```

The Apache RPMs should now be installed with the service running.

11.1.2 Test Apache

To verify that Apache is installed correctly, after it has been started, **open a web browser** and point it to the server. For example, the virtual server running on `LINUX1` can be reached with the following URL:

```
http://9.12.7.1/
```

You should see the following test page (Figure 11-1) to verify that the web server is working.

Figure 11-1 Apache test page

If you get an error in starting Apache, look in the `/var/log/httpd/error-log` log file for clues. If Apache started successfully but you cannot reach the test page from a browser, try accessing it using the IP address rather than the DNS name.

If you are fond of using a command-line interface, you can test the Apache configuration using **`lynx`**:

- To install lynx, perform the following step:

```
# yum -y install lynx
...
Installed:
  lynx.s390x 0:2.8.6-27.el6

Complete!
```

- To verify that Apache is installed correctly, after it has been started issue the following command in a shell environment:

```
# lynx http://9.12.7.1 -dump

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.
    ___________________________________________________________________
...
```

11.1.3 Turn on a firewall

RHEL 6.4 comes with an IP tables firewall. In section 9.2.3, "Turn off unneeded services" on page 185, it was recommended that you turn off the iptables service. If you did this on the golden image, the firewall is turned off on this clone. This section describes how to quickly enable an IP tables firewall and configure it to allow web traffic through. Perform the following steps:

1. Verify that the firewall is off with 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
```

2. 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  ]
```

3. Go back to your browser and click **refresh**. You should get an error that the server is not responding (or **Unable to connect**). This is because packets for ports for http: and https: (80 and 443) are dropped by default.

4. To allow web traffic through, modify the /etc/sysconfig/iptables file. First, make a copy of the original:

```
# cd /etc/sysconfig
# cp iptables iptables.orig
```

5. Add two rules (in bold) to allow these ports, then save your changes:

```
# vi iptables
# Firewall configuration written by system-config-firewall
# Manual customization of this file is not recommended.
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 443 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
```

6. Restart the firewall to pick up the new rules:

```
# service iptables restart
iptables: Flushing firewall rules: [  OK  ]
iptables: Setting chains to policy ACCEPT: filter [  OK  ]
iptables: Unloading modules: [  OK  ]
iptables: Applying firewall rules: [  OK  ]
```

7. 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 to pass through.

11.1.4 Configure SSL

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. Perform the following steps:

1. To use SSL, the `mod_ssl` package is required. Show that SSL communications *do not work* by changing `http` to **https** in the URL in your browser:

   ```
   https://9.12.7.1/
   ```

   You should see some type of communications error.

2. Install the `mod_ssl` RPM with the **`yum -y install`** command:

   ```
   # yum -y install mod_ssl
   ...
   Installed:
     mod_ssl.s390x 1:2.2.15-26.el6

   Complete!
   ```

3. Verify that the RPM was added:

   ```
   # rpm -qa | grep mod_ssl
   mod_ssl-2.2.15-26.el6.s390x
   ```

4. Restart the web server:

   ```
   # service httpd restart
   Stopping httpd: [  OK  ]
   Starting httpd: [  OK  ]
   ```

5. Go back to your browser and click **restart** again. This time you should get a warning about a self-signed certificate, which is acceptable for a test system. For a production website, you will probably want to obtain a certificate that is signed by a certificate authority (CA).

   To verify using lynx, perform the following command:

   ```
   # lynx https://9.12.7.1

   SSL error:self signed certificate-Continue? (y)
   SSL error:host(9.12.7.2)!=cert(virtcook2.itso.ibm.com)-Continue? (y)

   Test Page for the Apache HTTP Server on Red Hat Enterprise Linux (p1 of 2)
                          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.
        __________________________________________________________________
   ...
   ```

This section covered how to add SSL to your web server.

11.1.5 Populate the website

Web pages are stored in the `/var/www/html/` directory, which is the default web root. For example:

```
# cd /var/www/html/
# echo "<h2>Our new Web site</h2>" > index.html
```

Click refresh again and accept the certificate. You should see a web page similar to what is shown in Figure 11-2.

Figure 11-2 Testing the populating of your website

This section covered how to install and configure Apache as a web server.

11.2 Create an application development appliance

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 is software available on Linux to help form a development system for developers to create integrated applications. MySQL and Apache are among them. A popular open source web platform is LAMP, which stands for the open source software and programming languages used to make up the platform: Linux, Apache, MySQL, Python, or PHP.

In this example, the system running on `LINUX2` is used, which was installed by using kickstart. This will not have **yum** configured for online updates, so it was set up manually. See section 8.2.3, “Configure yum” on page 165.

1. Start an SSH session as `root` to the Linux system, which will be used.
2. Before installing the development tools, note how full the root and `/usr/` file systems are:

```
# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/dasda1          1008M  183M  774M  20% /
tmpfs                 246M     0  246M   0% /dev/shm
/dev/mapper/system_vg-opt_lv
                      504M   17M  462M   4% /opt
/dev/mapper/system_vg-tmp_lv
                      504M   17M  462M   4% /tmp
/dev/mapper/system_vg-usr_lv
                      2.0G  1.3G  614M  68% /usr
/dev/mapper/system_vg-var_lv
                      504M   95M  385M  20% /var
...
```

In this example, they are 20% and 68% full.

3. Use the **yum -y groupinstall** command to install the groups named **development-tools** and **development-libs**. This adds about 45 packages, which requires a number of minutes to install:

```
# yum -y groupinstall "Development tools" "Development libs"
...
Installed:
  autoconf.noarch 0:2.63-5.1.el6           automake.noarch 0:1.11.1-4.el6
  bison.s390x 0:2.4.1-5.el6                byacc.s390x 0:1.9.20070509-7.el6
  cscope.s390x 0:15.6-6.el6                ctags.s390x 0:5.8-2.el6
  diffstat.s390x 0:1.51-2.el6              doxygen.s390x 1:1.6.1-6.el6
  flex.s390x 0:2.5.35-8.el6                gcc.s390x 0:4.4.7-3.el6
  gcc-c++.s390x 0:4.4.7-3.el6              gcc-gfortran.s390x 0:4.4.7-3.el6
  git.s390x 0:1.7.1-2.el6_0.1              indent.s390x 0:2.2.10-7.el6
  intltool.noarch 0:0.41.0-1.1.el6         libtool.s390x 0:2.2.6-15.5.el6
  patchutils.s390x 0:0.3.1-3.1.el6         rcs.s390x 0:5.7-37.el6
  redhat-rpm-config.noarch 0:9.0.3-42.el6    rpm-build.s390x 0:4.8.0-32.el6
  subversion.s390x 0:1.6.11-7.el6          swig.s390x 0:1.3.40-6.el6
  systemtap.s390x 0:1.8-7.el6

Dependency Installed:
  cloog-ppl.s390x 0:0.15.7-1.2.el6      cpp.s390x 0:4.4.7-3.el6
  gettext-devel.s390x 0:0.17-16.el6     gettext-libs.s390x 0:0.17-16.el6
  glibc-devel.s390x 0:2.12-1.107.el6    glibc-headers.s390x 0:2.12-1.107.el6
  kernel-devel.s390x 0:2.6.32-358.el6   kernel-headers.s390x 0:2.6.32-358.el6
  libart_lgpl.s390x 0:2.3.20-5.1.el6    libgcj.s390x 0:4.4.7-3.el6
  libstdc++-devel.s390x 0:4.4.7-3.el6   mpfr.s390x 0:2.4.1-6.el6
  neon.s390x 0:0.29.3-2.el6             pakchois.s390x 0:0.4-3.2.el6
  perl-Error.noarch 1:0.17015-4.el6     perl-Git.noarch 0:1.7.1-2.el6_0.1
  ppl.s390x 0:0.10.2-11.el6             systemtap-client.s390x 0:1.8-7.el6
  systemtap-devel.s390x 0:1.8-7.el6

Complete!
```

4. Your application development server is now ready to use. You can choose to add or remove different packages.

5. Use the **df -h** command to show your file systems. In this example, the root file system was not changed, but /usr/ is now 82% full:

```
# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/dasda1          1008M  184M  774M  20% /
tmpfs                 246M     0  246M   0% /dev/shm
/dev/mapper/system_vg-opt_lv
                      504M   17M  462M   4% /opt
/dev/mapper/system_vg-tmp_lv
                      504M   17M  462M   4% /tmp
/dev/mapper/system_vg-usr_lv
                      2.0G  1.6G  363M  82% /usr
...
```

The Linux system should now have many application development tools installed and ready to use.

11.2.1 Configure subversion

If you need to set up a source code control system, subversion (or *svn)* is a good choice. To set up subversion, perform the following steps:

1. Verify that subversion was installed in the previous step:

```
# rpm -q subversion
subversion-1.6.11-7.el6.s390x
```

2. Set up an `svn` group and user with both the *gid* and *uid* being `500`:

```
# groupadd -g 500 svn
# useradd -c "SVN" -d /var/svnrepos -g 500 -u 500 -s /sbin/nologin -M svn
```

3. Install xinetd:

```
# yum -y install xinetd
...
Installed:
  xinetd.s390x 2:2.3.14-38.el6

Complete!
```

4. Set up an **`xinetd`** configuration file for subversion:

```
# cd /etc/xinetd.d/
# vi svn
# description: svnserve allows access to Subversion repositories using
#              the svn network protocol.
service svn
{
        disable                 = no
        port                    = 3690
        socket_type             = stream
        protocol                = tcp
        wait                    = no
        user                    = svn
        server                  = /usr/bin/svnserve
        server_args             = -i -r /var/svnrepos
}
```

   The `-r /var/svnrepos` parameter means that this is the starting directory when a client starts a session with the server.

5. Start **`xinetd`**:

```
# service xinetd start
Starting xinetd: [  OK  ]
```

6. Check that the server is now listening with the **`netstat -l`** command:

```
# netstat -l | grep svn
tcp        0      0 *:svn                       *:*                         LISTEN
```

7. Make a directory for the subversion repositories and change into that directory:

```
# mkdir /var/svnrepos
# cd /var/svnrepos
```

8. Create a new repository (**myrepo** in this example) with the **`svnadmin create`** command:

```
# svnadmin create myrepo
# cd myrepo
```

9. Make the `svn` user and group owner of all new directories:

```
# chown -R svn.svn /var/svnrepos
```

10. Set up the configuration file for the new project for local access. In the `conf/svnserve.conf` file, uncomment and set **anon-access** to none, and uncomment the **auth-access** and **passwd-db** lines:

```
# cd conf
# vi svnserve.conf
### This file controls the configuration of the svnserve daemon, if you
### use it to allow access to this repository.  (If you only allow
### access through http: and/or file: URLs, then this file is
### irrelevant.)

### Visit http://subversion.tigris.org/ for more information.

[general]
### These options control access to the repository for unauthenticated
### and authenticated users.  Valid values are "write", "read",
### and "none".  The sample settings below are the defaults.
anon-access = none
auth-access = write
### The password-db option controls the location of the password
### database file.  Unless you specify a path starting with a /,
### the file's location is relative to the directory containing
### this configuration file.
### If SASL is enabled (see below), this file will NOT be used.
### Uncomment the line below to use the default password file.
password-db = passwd
...
```

11. Add a user and password. In this example it is **fmiranda** and **passwd**:

```
# vi passwd
#### This file is an example password file for svnserve.
### Its format is similar to that of svnserve.conf. As shown in the
### example below it contains one section labelled [users].
### The name and password for each user follow, one account per line.

[users]
...
fmiranda = passwd
```

The subversion server should now be configured with one empty project, `myrepo` in this example.

11.2.2 Additional resources

The following websites are resources for additional information about application development topics:

Scripting languages

- http://www.perl.com
- http://www.python.org
- http://www.freeos.com/guides/lsst

C/C++

- http://gcc.gnu.org/onlinedocs/gcc
- http://en.wikipedia.org/wiki/GNU_Compiler_Collection#External_links
- http://www.cs.umd.edu/class/spring2002/cmsc214/Tutorial/makefile.html
- 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

11.3 Create an LDAP server appliance

The Lightweight Directory Access Protocol (LDAP) is commonly implemented with the OpenLDAP package, which comes standard with most Linux distributions. Among other directory functions, OpenLDAP allows for centralized login authentication and user and group ID resolution.

The LDAP server is installed on the virtual machine that was configured to be a web server, `LINUX1` in this example. Later, the application development appliance that was just created is set to use the new LDAP server.

The following steps are covered in this section:

1. "Install the OpenLDAP server"
2. "Configure the OpenLDAP server" on page 215
3. "Start the LDAP service" on page 216
4. "Add an LDAP user" on page 216
5. "Configure an LDAP client" on page 218

11.3.1 Install the OpenLDAP server

Perform the following steps to create an OpenLDAP server:

1. Start an SSH session as `root` on the Linux system where Apache was installed in section 11.1, "Create a web server appliance" on page 206. If you have not performed those steps, they should be done now because Apache will be needed. In this example, it is running on the virtual machine `LINUX1`.
2. Use the **yum** command to install the OpenLDAP client and server RPMs:

```
# yum -y install openldap-clients openldap-servers
...
Installed:
  openldap-clients.s390x 0:2.4.23-31.el6 openldap-servers.s390x 0:2.4.23-31.el6

Dependency Installed:
  libtool-ltdl.s390x 0:2.2.6-15.5.el6

Complete!
```

OpenLDAP is now installed.

11.3.2 Configure 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:

1. 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. In this example, ***itso.ibm.com*** is used.
2. The LDAP administrator or root name and password.

Perform the following steps:

1. 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 might want to make a copy of it.

```
# slappasswd
New password: lnx4vm
Re-enter new password: lnx4vm
{SSHA}th6d+bfXgap5N+Pzf97AewivW4FWEXsO
```

2. The OpenLDAP server configuration file that will contain the LDAP manager (root) password is `/etc/openldap/slapd.d/cn=config/olcDatabase={1}bdb.ldif`. Make a backup copy of that file:

```
# cd /etc/openldap/slapd.d/cn=config
# cp olcDatabase={2}bdb.ldif olcDatabase={2}bdb.ldif.orig
```

3. Edit the file and change three lines:
   - The `olcSuffix` to a value based on your organization's DNS domain name. This example corresponds to `itso.ibm.com`.
   - Add one line to set the LDAP manager's password. Use the variable `olcRootPW` and set the password to the output of the previous **slappasswd** command:

```
# vi olcDatabase={2}bdb.ldif
dn: olcDatabase={2}bdb
objectClass: olcDatabaseConfig
objectClass: olcBdbConfig
olcDatabase: {2}bdb
olcSuffix: dc=itso,dc=ibm,dc=com
olcAddContentAcl: FALSE
olcLastMod: TRUE
olcMaxDerefDepth: 15
olcReadOnly: FALSE
olcRootDN: cn=Manager,dc=itso,dc=ibm,dc=com
olcRootPW: {SSHA}th6d+bfXgap5N+Pzf97AewivW4FWEXsO
olcSyncUseSubentry: FALSE
olcMonitoring: TRUE
olcDbDirectory: /var/lib/ldap
...
```

Your LDAP server is now minimally configured.

11.3.3 Start the LDAP service

To start the LDAP server, perform the following steps:

1. Start LDAP at boot time with the **chkconfig** command and for this session with the **service** command:

```
# chkconfig slapd on
# service slapd start
Starting slapd: [  OK  ]
```

2. Query the LDAP database with the **ldapsearch** command. The **-x** flag specifies that simple authentication is used:

```
# ldapsearch -x
# extended LDIF
#...
# 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 because no data has been added to it.

11.3.4 Add an LDAP user

Define a new LDAP user. To do so, perform the following steps:

1. Choose an LDAP user name. In this example, fmiranda will be used. Verify that there is no such local user with the **id** command:

```
# id fmiranda
id: fmiranda: No such user
```

2. An LDIF (LDAP Interchange Format) file is created to add an organizational unit named **People** and a user named **fmiranda**. Create a similar file for your system's values:

```
# cd /tmp
# vi initial.ldif // create the input file ...
dn: dc=itso,dc=ibm,dc=com
objectClass: dcObject
objectClass: organization
description: itso.ibm.com domain
o: itso
dc: itso

dn: cn=Manager,dc=itso,dc=ibm,dc=com
objectClass: organizationalRole
cn: Manager

dn: ou=People,dc=itso,dc=ibm,dc=com
ou: People
objectClass: top
objectClass: organizationalUnit

dn: uid=fmiranda,ou=People,dc=itso,dc=ibm,dc=com
uid: fmiranda
cn: fmiranda
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
```

```
loginShell: /bin/bash
uidNumber: 10000
gidNumber: 10000
homeDirectory: /home/fmiranda

dn: ou=Group,dc=itso,dc=ibm,dc=com
objectClass: top
objectClass: organizationalUnit
ou: Group

dn: cn=fmiranda,ou=Group,dc=itso,dc=ibm,dc=com
objectClass: posixGroup
objectClass: top
cn: fmiranda
userPassword: {crypt}x
gidNumber: 10000
```

3. Add the contents of the LDIF file to the LDAP server with the **ldapadd** command:

```
# ldapadd -x -h localhost -D "cn=Manager,dc=itso,dc=ibm,dc=com" -f initial.ldif -W
Enter LDAP Password: lnx4vm
adding new entry "dc=itso,dc=ibm,dc=com"

adding new entry "cn=Manager,dc=itso,dc=ibm,dc=com"

adding new entry "ou=People,dc=itso,dc=ibm,dc=com"

adding new entry "uid=fmiranda,ou=People,dc=itso,dc=ibm,dc=com"

adding new entry "ou=Group,dc=itso,dc=ibm,dc=com"

adding new entry "cn=fmiranda,ou=Group,dc=itso,dc=ibm,dc=com"
```

4. Set the base distinguished name to **dc=itso,dc=ibm,dc=com**. This is set in the BASE variable in the LDAP client configuration file /etc/openldap/ldap.conf:

```
# cd /etc/openldap
# cp ldap.conf ldap.conf.orig
# vi ldap.conf
#
# LDAP Defaults
#

# See ldap.conf(5) for details
# This file should be world readable but not world writable.

BASE dc=itso,dc=ibm,dc=com
...
```

5. Search for the new user just added with the **ldapsearch** command:

```
# ldapsearch -x uid=fmiranda

# extended LDIF
...
# fmiranda, Peoplee, itso.ibm.com
dn: uid=fmiranda,ou=People,dc=itso,dc=ibm,dc=com
uid: fmiranda
cn: fmiranda
objectClass: account
objectClass: posixAccount
objectClass: top
```

```
objectClass: shadowAccount
loginShell: /bin/bash
uidNumber: 10000
gidNumber: 10000
homeDirectory: /home/fmiranda

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

This shows that the user exists in the LDAP database and that LDAP is working.

6. Set the password of the new user with the **ldappasswd** command. You will also need to provide the LDAP administrator password:

```
# ldappasswd -x -D "cn=Manager,dc=itso,dc=ibm,dc=com" -W -S
"uid=fmiranda,ou=People,dc=itso,dc=ibm,dc=com"
New password: lnx4vm
Re-enter new password: lnx4vm
Enter LDAP Password: lnx4vm
```

7. Create a home directory with the **mkdir** command:

```
# mkdir /home/fmiranda
```

You have now added a new LDAP user using an LDIF file, and have set the new LDAP user's password. You should now have the OpenLDAP server installed, configured, and populated.

11.3.5 Configure an LDAP client

You are now ready to configure a system to authenticate using the new LDAP server. In this example, the Linux system running on the LINUX2 virtual machine is configured as an LDAP client. To do so, perform the following steps:

1. Start an SSH session as `root` to the Linux system you want to set up as an LDAP client.
2. Verify that you have the `sssd` RPMs installed with the following command:

```
# rpm -qa | grep sssd
sssd-client-1.9.2-82.el6.s390x
sssd-1.9.2-82.el6.s390x
```

3. Verify that the value of `FORCELEGACY` in the `/etc/sysconfig/authconfig` file is set to `no`:

```
# grep FORCELEGACY /etc/sysconfig/authconfig
FORCELEGACY=no
```

This setting will ensure the new recommended style of authentication will be used.

4. Invoke the **authconfig-tui** command. Use the **Tab** key to move between fields, the **space bar** to change selections and the **Enter** key to select. Set the **Use LDAP** under *User Information*, and **Use LDAP Authentication** under *Authentication*. Move to **Next** and press **Enter**:

```
# 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 Fingerprint reader           ¦
¦                         [ ] Use Winbind Authentication       ¦
¦                         [*] Local authorization is sufficient ¦
¦                                                              ¦
¦          ----------                      --------            ¦
¦          ¦ Cancel ¦                      ¦ Next ¦            ¦
¦          ----------                      --------            ¦
¦                                                              ¦
¦                                                              ¦
---------------------------------------------------------------¦
```

5. On the next screen, set the *Server* value to point to the LDAP server. In this example, it is **ldap://*virtcook1.itso.ibm.com/***. Set the `Base DN` to your LDAP suffix value. In this example it is ***dc=itso,dc=ibm,dc=com***. Select **OK**.

```
------------------- LDAP Settings -------------------¦
¦                                                    ¦
¦          [ ] Use TLS                               ¦
¦  Server: ldap://virtcook1.itso.ibm.com/__________  ¦
¦ Base DN: dc=itso,dc=ibm,dc=com__________________   ¦
¦                                                    ¦
¦         --------                 ------            ¦
¦         ¦ Back ¦                 ¦ Ok ¦            ¦
¦         --------                 ------            ¦
¦                                                    ¦
¦                                                    ¦
-----------------------------------------------------¦
```

6. You might get a warning about a certificate. If so, select **OK**.
7. The LDAP client (in this example on `LINUX2`) should now be pointing to the LDAP server (in this example on `LINUX1`). Test it with the **`id`** command:

```
# id fmiranda
uid=10000(fmiranda) gid=10000(fmiranda) groups=10000(fmiranda)
```

   This shows that the LDAP client is now communicating with the LDAP server.

In the recent version of RHEL 6, it is recommended that you authenticate over SSH using TLS/SSL, which encrypts the traffic over the network. So the next step is to set that up.

11.3.6 Setting up TLS/SSL on OpenLDAP

Setting up TLS/SSL is recommended so that LDAP traffic is encrypted.

Note: The following steps were based on Red Hat articles 65603 and15497 about how to set up LDAP for TLS/SSL.

Set up a certificate authority

To set up LDAP TLS/SSL with `/etc/pki/CA` as the certificate authority's working directory, perform the following steps:

1. Start a session as `root` on the system where the LDAP server was configured (in this example, it is `LINUX1`).
2. Change directory to `/etc/pki/CA/private/`:

```
# cd /etc/pki/CA/private
```

3. Generate a private key and a self-signed CA certificate. You will be prompted for a pass phrase, which will be needed later. The parentheses in the following command create a subshell where both of the following commands are run:

```
# (umask 077; openssl genrsa -out cakey.pem -aes256 4096)

Generating RSA private key, 4096 bit long modulus
.....................................................................................
......................++
............................................++
e is 65537 (0x10001)
Enter pass phrase for cakey.pem: lnx4vm
Verifying - Enter pass phrase for cakey.pem: lnx4vm
```

4. Show the private key file:

```
# ls -l cakey.pem
-rw-------. 1 root root 3326 Sep 14 06:15 cakey.pem
```

   Note there are read permissions only for the owner, root. This shows that the previous **umask** command took effect.

5. Create a certificate file. The number of days sets the certificate expiration to approximately six years. The `Common Name` field is set to `itso.ibm.com` certificate authority because the main CA will use just `itso.ibm.com` later.

```
# cd /etc/pki/CA
# openssl req -new -x509 -key private/cakey.pem -days 2190 > cacert.pem
Enter pass phrase for private/cakey.pem: lnx4vm
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:US
State or Province Name (full name) []:New York
Locality Name (eg, city) [Default City]:Poughkeepsie
Organization Name (eg, company) [Default Company Ltd]:IBM
Organizational Unit Name (eg, section) []:ITSO
Common Name (eg, your name or your server's hostname) []:virtcook1.itso.ibm.com
Certificate Authority
Email Address []:sysadmin@virtcook1.itso.ibm.com
```

6. Create a directory for HTTP certificates:

```
# mkdir /var/www/html/certs
```

7. *Publish* your CA certificate through HTTP.

```
# cp cacert.pem /var/www/html/certs
```

8. Create an empty file named `index.txt` that will be the certificate index:

```
# cd /etc/pki/CA
# touch index.txt
```

9. Create a file named `serial` to indicate the next certificate serial number to be issued:

```
# echo 01 > serial
```

10. Generate the LDAP server's certificate with more than 1024 bits in the key:

```
# openssl genrsa 2048 > /etc/openldap/certs/slapd_key.pem
Generating RSA private key, 2048 bit long modulus
..................................................+++
..........+++
```

```
e is 65537 (0x10001)
```

11.Create the certificate signing request file (note that there are two lines, but this is all one long command). No `challenge password` nor `optional company name` are required:

```
# openssl req -new -key /etc/openldap/certs/slapd_key.pem -out
/etc/openldap/certs/slapd.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:US
State or Province Name (full name) []:New York
Locality Name (eg, city) [Default City]:Poughkeepsie
Organization Name (eg, company) [Default Company Ltd]:IBM
Organizational Unit Name (eg, section) []:ITSO
Common Name (eg, your name or your server's hostname) []:virtcook1.itso.ibm.com
Email Address []:sysadmin@virtcook1.itso.ibm.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

The certificate signing request file is to be sent to the certificate authority if you are not creating a self-signed certificate.

12.Create the certificate signing request file:

```
# openssl ca -config /etc/pki/tls/openssl.cnf -out /etc/openldap/certs/slapd_cert.pem
-infiles /etc/openldap/certs/slapd.csr
Using configuration from /etc/pki/tls/openssl.cnf
Enter pass phrase for /etc/pki/CA/private/cakey.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 1 (0x1)
        Validity
            Not Before: Sep 14 12:33:47 2013 GMT
            Not After : Sep 14 12:33:47 2014 GMT
        Subject:
            countryName               = US
            stateOrProvinceName       = New York
            organizationName          = IBM
            organizationalUnitName    = ITSO
            commonName                = virtcook1.itso.ibm.com
            emailAddress              = sysadmin@virtcook1.itso.ibm.com
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                69:84:1F:BF:18:3A:AD:71:AE:D6:1A:5F:70:4A:81:D2:AE:54:29:83
            X509v3 Authority Key Identifier:
                keyid:26:4D:EB:7C:5C:D2:EB:FA:B6:8E:CA:58:46:D8:02:DA:0E:B9:21:51

Certificate is to be certified until Sep 14 12:33:47 2014 GMT (365 days)

Sign the certificate? [y/n]:y
```

```
1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

13. Make the /etc/openldap/cacerts/ directory:

```
# mkdir /etc/openldap/cacerts
```

14. Copy the CA certificate to the /etc/openldap/certs/ directory for the slapd server:

```
# cp /etc/pki/CA/cacert.pem /etc/openldap/cacerts
```

15. Run the **cacertdir_rehash** command to create hash symbolic links of certificate files:

```
# cacertdir_rehash /etc/openldap/cacerts
```

16. Run the **restorecon** command to reset SE Linux:

```
# restorecon -R /etc/openldap/certs
```

17. Reset the ownership and permission bits of files:

```
# chmod 600 /etc/openldap/certs/slapd_key.pem
# chown ldap:ldap /etc/openldap/certs/slapd*
```

18. Show the resulting files:

```
# ls -l /etc/openldap/certs/
total 16
-rw-r--r--. 1 ldap ldap 5955 Sep 14 08:33 slapd_cert.pem
-rw-r--r--. 1 ldap ldap 1090 Sep 14 08:33 slapd.csr
-rw-------. 1 ldap ldap 1675 Sep 14 08:32 slapd_key.pem
```

19. Stop LDAP with the **service** command:

```
# service slapd stop
Stopping slapd: [  OK  ]
```

20. Make a copy of the original file /etc/openldap/slapd.d/cn=config.ldif:

```
# cd /etc/openldap/slapd.d
# cp cn=config.ldif cn=config.ldif.orig
```

21. In the cn-config.ldif file, modify two lines and add two lines at the bottom:

```
# vi cn=config.ldif
...
olcTLSCertificateFile: /etc/openldap/certs/slapd_cert.pem
olcTLSCertificateKeyFile: /etc/openldap/certs/slapd_key.pem
olcTLSVerifyClient: never
olcToolThreads: 1
olcWriteTimeout: 0
structuralObjectClass: olcGlobal
entryUUID: 156e3d0c-add0-1032-9bc0-8dcb1bd7381d
creatorsName: cn=config
createTimestamp: 20130909191633Z
entryCSN: 20130909191633.488131Z#000000#000#000000
modifiersName: cn=config
modifyTimestamp: 20130909191633Z
olcTLSCipherSuite: HIGH:+SSLv3
olcTLSCACertificateFile: /etc/openldap/cacerts/cacert.pem
```

22. Edit the /etc/sysconfig/ldap file and set the SLAPD_LDAPS variable to **yes**:

```
# cd /etc/sysconfig
# vi ldap
...
# Run slapd with -h "... ldaps:/// ..."
#   yes/no, default: no
```

```
SLAPD_LDAPS=yes
...
```

23. Start LDAP with the **service** command:

```
# service slapd start
Starting slapd: [  OK  ]
```

Installing certificate on clients

The clients require the CA certificate to trust the server certificates signed by this CA, for example, LDAP clients expect the CA certificate to be present in the /etc/openldap/cacerts directory. If Apache server is configured to use SSL using the CA signed cert, the CA certificate has to be imported to the web browser:

- Start an SSH session as root to the LDAP client. In this example, it is virtcook2 running on LINUX2.
- Get the LDAP certificate with the **wget** command:

```
# wget http://virtcook1.itso.ibm.com/certs/cacert.pem -O
/etc/openldap/cacerts/cacert.pem
Resolving virtcook1.itso.ibm.com... 9.12.7.1
Connecting to virtcook1.itso.ibm.com|9.12.7.1|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2183 (2.1K) [text/plain]
Saving to: â/etc/openldap/cacerts/cacert.pemâ

100%[===========================================>] 2,183       --.-K/s   in 0s

2013-09-14 08:44:24 (290 MB/s) - â/etc/openldap/cacerts/cacert.pemâ

# cacertdir_rehash /etc/openldap/cacerts
```

The LDAP client now has the server's certificate in the /etc/openldap/cacerts/ directory.

Configure LDAP server for TLS/SSL

To further configure the LDAP server to use the new certificate, perform the following steps:

1. Start (or go back to) an SSH session as root to the LDAP server. In this example, it is virtcook1 running on LINUX1.
2. Create a new LDIF file that adds the server certificate, server's private key, and CA certificate:

```
# cd /tmp
# vi ldap_tls.ldif
add: olcTLSCACertificateFile
olcTLSCACertificateFile: /etc/openldap/cacerts/cacert.pem
add: olcTLSCertificateFile
olcTLSCertificateFile: /etc/openldap/certs/slapd_cert.pem
add: olcTLSCertificateKeyFile
olcTLSCertificateKeyFile: /etc/openldap/slapd_key.pem
```

3. Use the **ldapmodify** command to add the new certificate files to the OpenLDAP configuration:

```
# ldapmodify -x -f ldap_tls.ldif -h 9.12.7.1 -D cn=Manager,dc=itso,dc=ibm,dc=com -W
Enter LDAP Password:
```

4. Restart slapd service:

```
# service slapd restart
Stopping slapd: [  OK  ]
Starting slapd: [  OK  ]
```

5. Test the SSL/TLS configuration by omitting the -x flag (use simple authentication) on the **ldapsearch** command:

```
# ldapsearch uid=fmiranda
SASL/GSSAPI authentication started
ldap_sasl_interactive_bind_s: Local error (-2)
        additional info: SASL(-1): generic failure: GSSAPI Error: Unspecified GSS
failure.  Minor code may provide more information (Credentials cache file
'/tmp/krb5cc_0' not found)
```

This section shows SSL/TLS working on the LDAP server.

Set the LDAP client to use TLS/SSL

Test the configuration by setting the LDAP client to use TLS/SSL over the wire.

Perform the following steps:

1. Start an SSH session as root to the LDAP client. In this example, it is the Linux system running on ***LINUX2***.
2. Copy the certificate from the `/etc/pki/CA/private/` directory to:

```
# cd /etc/openldap/cacerts
# scp virtcook1:/etc/pki/CA/private/my-ca.crt .
my-ca.crt                                          100% 1428     1.4KB/s   00:00
```

3. Set the system to use SSL/TLS on the second panel of the **authconfig-tui** tool by selecting the Use TLS check box:

```
# authconfig-tui
...
                +-------------------¤ LDAP Settings |----------------+
                |                                                    |
                |          [*] Use TLS                               |
                |  Server: ldap://virtcook1/_______________________  |
                | Base DN: dc=itso,dc=ibm,dc=com___________________  |
                |                                                    |
                |         |------|                 |----|            |
                |         | Back |                 | Ok |            |
                |         |------|                 |----|            |
                |                                                    |
                |                                                    |
                +----------------------------------------------------+
```

4. Try starting an SSH session on the LDAP client as an LDAP user:

```
# ssh fmiranda@localhost
The authenticity of host 'localhost (::1)' can't be established.
RSA key fingerprint is 6a:cc:5e:30:da:84:11:03:b1:d4:a3:aa:4d:c2:e0:53.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'localhost' (RSA) to the list of known hosts.
fmiranda@localhost's password:
...
```

5. Try to SSH in to the LDAP client (`virtcook2`) as the LDAP user, ***fmiranda*** in this example:

```
login as: fmiranda
fmiranda@9.12.7.2's password: lnx4vm
-bash-4.1$
```

Being able to get an SSH session shows that the LDAP client and server are now configured to use TLS/SSL.

More details about the `cn=config/` directory replacing the `/etc/openldap/slapd.conf` file is on the following Red Hat website (you might need a subscription to get to it):

https://access.redhat.com/kb/docs/DOC-3637

11.4 Create a file server appliance

Samba allows MS Windows clients to map Linux file systems as shared drives. Samba can also act as a middle tier between Windows clients and a Linux print server. The recommended Linux print server is the Common UNIX Printing System (CUPS). This section does not describe the configuration of CUPS but it does describe how the necessary RPMs are installed.

The steps in this section are as follow:

1. "Install necessary RPMs"
2. "Configure Samba configuration file" on page 225
3. "Adding a Samba user" on page 226
4. "Start Samba at boot time" on page 227
5. "Test your changes" on page 227

11.4.1 Install necessary RPMs

To install the Samba RPMs, perform the following steps:

1. Start an SSH session to the Linux system onto which you will install Samba. In this example, it is `LINUX2`.
2. Add the following RPM with the **`yum -y`** command:

```
# yum -y install samba
...
Installed:
  samba.s390x 0:3.6.9-151.el6

Complete!
```

3. Confirm that the RPMs were added:

```
# rpm -qa | grep samba
samba4-libs-4.0.0-55.el6.rc4.s390x
samba-common-3.6.9-151.el6.s390x
samba-3.6.9-151.el6.s390x
samba-winbind-clients-3.6.9-151.el6.s390x
samba-winbind-3.6.9-151.el6.s390x
samba-client-3.6.9-151.el6.s390x
```

11.4.2 Configure Samba configuration file

The one configuration file for Samba is `/etc/samba/smb.conf`. It is easy to add an SMB share that will be made available by the Samba server. A good test directory is `/usr/share/doc/` because it has excellent Linux documentation. The following example will create a file *share* named `sharedoc`.

To do so, perform the following steps:

1. Change directory to `/etc/samba/` and back up the configuration file:

```
# cd /etc/samba
```

```
# cp smb.conf smb.conf.orig
```

2. Add three lines at the bottom of the file defining the *share* named `sharedoc`:

```
# vi smb.conf    // add three lines at the bottom of the file:
...
[sharedoc]
  comment = RHEL 6.4 on System z documentation
  path = /usr/share/doc/
```

3. Verify the syntax of your changes with the **testparm** command. You should see a reference to the new `sharedoc` section that was just added:

```
# 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 `/usr/share/doc` directory and below.

11.4.3 Adding a Samba user

The default method that Samba uses to determine users' credentials is to look in the `/etc/samba/smbpasswd` file. That user must first exist in the Linux file system (`/etc/passwd`, `/etc/shadow`, and so on).

This method of maintaining Samba users, groups, and passwords is good for a few users. For many users, merging Samba and LDAP is recommended. It is not as simple as pointing the virtual file and print server at the virtual LDAP server as described in 11.3, "Create an LDAP server appliance" on page 214 because the Samba schema must first be added to LDAP. Details are outside the scope of this book.

To add a Samba user, perform the following steps:

1. 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
BAD PASSWORD: it is based on a dictionary word
BAD PASSWORD: is too simple
Retype new password: lnx4vm
passwd: all authentication tokens updated successfully.
```

2. Add the user `sambauser1` to the `smbpasswd` file with 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.
```

The local and Samba user `sambauser1` should now be added to the system.

11.4.4 Start Samba at boot time

Samba can be started for the current session with the **`service`** command and at boot time with the **`chkconfig`** command. Do this for both the **`smb`** and **`nmb`** services:

```
# service smb start
Starting SMB services: [  OK  ]
# service nmb start
Starting NMB services: [  OK  ]
# chkconfig smb on
# chkconfig nmb on
```

Samba should now be running and configured to start at boot time.

11.4.5 Test your changes

Verify the configuration settings with the following steps:

1. Verify that Samba, which is composed of the **`smb`** and **`nmb`** services, is running with the following **`service`** command:

```
# service smb status
smbd (pid 5891) is running...
# service nmb status
nmbd (pid 5905) is running...
```

2. Verify the shares that are available with the following **`smbclient`** command:

```
# smbclient -U sambauser1 -L localhost
Enter sambauser1's password:
Domain=[MYGROUP] OS=[Unix] Server=[Samba 3.6.9-151.el6]

        Sharename       Type      Comment
        ---------       ----      -------
        sharedoc        Disk      RHEL 6.4 on System z documentation
        IPC$            IPC       IPC Service (Samba Server Version 3.5.6-86.el6)
        sambauser1      Disk      Home Directories
...
```

To test getting a Samba share from a Windows desktop, perform the following steps:

1. Go to any MS Windows Explorer window (such as *My Computer*) and select **Tools → Map Network Drive**.
2. Use the Universal Naming Convention (UNC) to specify the Samba server and share the name, as shown in the upper left corner of Figure 11-3 on page 228. In this example, the UNC is **`\\`*`9.12.7.2`*`\sharedoc`**.
3. You might have to click **different user name** if the user or password on the new Samba server is different from the Windows system you are connecting from.
4. Click **Finish**.

If all the steps were correct, you should see the files in a new Explorer window, as shown in the lower right corner of Figure 11-3.

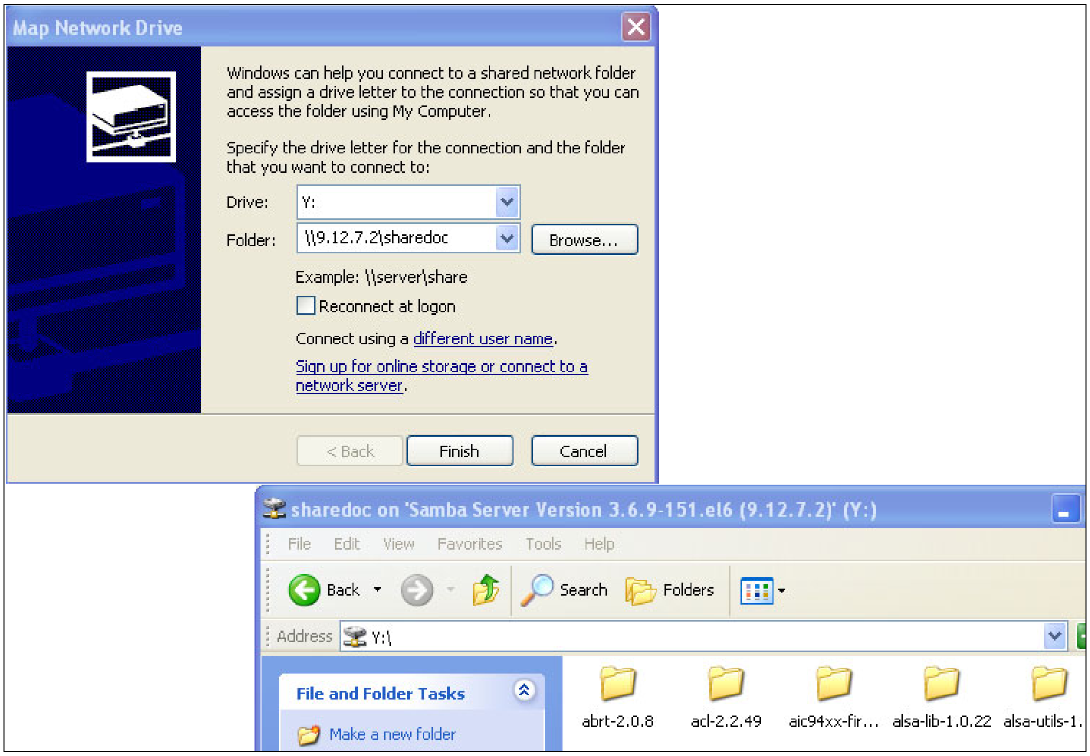


Figure 11-3 Mapping a network drive to the Samba server

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 with the following **net use** command:

```
c:\>net use y: \\9.12.7.2\sharedoc

Enter the user name for '9.12.7.2': sambauser1
Enter the password for 9.12.7.2:
The command completed successfully.
```

Detach the share with the following **net use** command:

```
c:\>net use y: /delete
y: was deleted successfully.
```

11.4.6 Configure printing

Configuring printing is beyond the scope of this section. For details, see the IBM Redpaper publication *Printing with Linux on zSeries Using CUPS and Samba*, REDP-3864, at the following website:

http://www.redbooks.ibm.com/abstracts/redp3864.html

Congratulations. You should now have two virtual servers set up with Apache and application development tools on the first virtual server, and LDAP and Samba on the second.

12

Installing Red Hat Enterprise Linux with kickstart

"*We still don't know one thousandth of one percent of what nature has revealed to us.*"

— Albert Einstein

Kickstart is an automated way of installing RHEL 6.4. By using kickstart, you can create a single file that answers all of the questions that are usually asked during an interactive installation.

In the previous chapter, you cloned to `RH64GOLD` and created two new virtual machines for virtual servers. In this chapter, you *kickstart* an RHEL 6.4 system to `LINUX3`. In comparison, cloning a server is faster, assuming the `FLASHCOPY` command is available. However, kickstart-ing a server is more flexible because it allows for different package configurations as well as preinstallation and postinstallation scripting.

The Linux administration system is now configured as an installation server using NFS to share the installation tree. Configure it as a kickstart server to perform automated installations over the network. The following steps are involved in installing Linux with kickstart:

- Configure the Linux administration system for kickstart
- "Configure LINUX1 for kickstart" on page 231
- "Kickstart the LINUX1 user" on page 233

12.1 Configure the Linux administration system for kickstart

The installer generates a kickstart file at the end of every installation. It is based on the answers provided during the interactive installation. This kickstart file is named `anaconda-ks.cfg` and is in the `/root/` directory. This file on `LNXADMIN` will be used as a template for `LINUX3`.

Perform the following steps:

1. Start an SSH session on the Linux administration system (`LNXADMIN`) as root.

2. Start the golden image on `RH64GOLD`. You could log on to a 3270 session, but you can also start it from the Linux administration system with the CP **`XAUTOLOG`** command:

```
# vmcp xautolog rh64gold
Command accepted
```

3. Create the /var/`nfs/ks/` directory for the kickstart file:

```
# cd /var/nfs
# mkdir ks
# cd ks
```

4. Copy the sample kickstart file from the golden image:

```
# scp 9.12.7.9:/root/anaconda-ks.cfg linux3-ks.cfg
anaconda-ks.cfg                               100% 1813     1.8KB/s   00:00
```

5. Set the read bits on the file with the **`chmod +r`** command:

```
# chmod +r linux3-ks.cfg
```

6. Edit the kickstart configuration file as follows. After the first four changes, which are in bold, remove the comments from the part, `volgroup` and `logvol` lines. Edit the lines in bold to customize this kickstart for `LINUX1`:

```
# vi linux3-ks.cfg
# Kickstart file automatically generated by anaconda.

#version=RHEL6
install
reboot
nfs --server=9.12.7.7 --dir=/var/nfs/rhel64
lang en_US.UTF-8
rootpw  --iscrypted
$6$jiFGqyU1FwxWWQ6t$7qnsOSsUsNOyGnjtIpR63z2O4RDjL1q6M//lxfA.E5SbQ.M2gNKCJpahQ.mO7JCm.56y
H3vKbxc5bVtvRERwdO
firewall --disabled
authconfig --enableshadow --passalgo=sha512 --enablefingerprint
selinux --enforcing
timezone --utc America/New_York
bootloader --location=mbr --driveorder=dasdb,dasdc,dasdd,dasde
--append="crashkernel=auto"
# 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
clearpart --all --initlabel --drives=dasdb,dasdc,dasdd,dasde
part / --fstype=ext4 --size=1024
part pv.094006 --grow --size=200
part pv.094009 --grow --size=200
part swap --grow --size=200
part swap --grow --size=200
volgroup system_vg --pesize=4096 pv.094006 pv.094009
logvol /opt --fstype=ext4 --name=opt_lv --vgname=system_vg --size=512
logvol /tmp --fstype=ext4 --name=tmp_lv --vgname=system_vg --size=512
logvol /usr --fstype=ext4 --name=usr_lv --vgname=system_vg --size=2048
logvol /var --fstype=ext4 --name=var_lv --vgname=system_vg --size=512

%packages
@base
...
%end
```

The following list provides clarification to some of the values:

– The line `reboot` is added to set the server to automatically shut down after kickstart.
– The line starting with `nfs --server=` sets the IP address of the installation server and path to the installation tree.
– The line starting with `firewall` disables the firewall. This is not recommended if the server is on an external network.
– The line starting with `bootloader` removes references to additional drives that are only available to the Linux administration system.
– The line starting with zerombr is required to initialize disks with invalid partition tables. During guest start-up, the 300-301 VDISKs are created uninitialized.
– The line starting with `clearpart --all` specifies to remove all existing partitions.
– The line starting with `part /` defines the root partition to be 512 MB of type ext4.
– The line starting with `part swap` defines a swap partition of size 512 MB.
– The two lines starting with `part pv` specify to make physical volumes.
– The next two lines starting with `part swap` define partitions. Because they have the `--grow` parameter, all of the virtual disks will be used for swap, regardless of the size specified. Anaconda creates the swap devices based on the order in the kickstart file, so the swap spaces will be created on virtual disks `300` and `301`.
– The line starting with `volgroup` specifies to create a volume group.
– The next four lines starting with `logvol` define the logical volumes created when installing.
– The line `@base` specifies a default set of packages for the installation. These can be customized later by adding or removing specific packages from the `%packages` section.

7. Add the path to the kickstart folder to `/etc/exports`:

```
# vi /etc/exports
/var/nfs/rhel64/            *(ro,sync)
/var/nfs/SG248147           *(ro,sync)
/var/nfs/ks                 *(ro,sync)
```

8. Restart the NFS service on the Linux administration system. The **`showmount -e`** command should show the exported file systems:

```
# service nfs reload
# showmount -e
/var/nfs/ks        *
/var/nfs/SG248147  *
/var/nfs/rhel64    *
```

You should now be ready to configure kickstart for a virtual machine.

12.2 Configure LINUX1 for kickstart

You should have created a virtual machine `LINUX1`. If you did not, see section 10.1, “Define three new virtual machines” on page 194.

The two Linux minidisks, `100` and `101`, must be formatted for Linux. To do so, perform the following steps:

1. Start an SSH session as `root` to the Linux administration system, ***`virtcook7`*** in this example.

2. Link the target disks, LINUX1 100 and 101, multi-read (read/write if no other virtual machine has write access) as virtual devices 2100 and 2101:

```
# vmcp link linux1 100 2100 mr
# vmcp link linux1 101 2101 mr
```

3. Bring the devices online so they can be formatted:

```
# chccwdev -e 2100-2101
Setting device 0.0.2100 online
Done
Setting device 0.0.2101 online
Done
```

4. Verify that the minidisks are online:

```
# lsdasd
Bus-ID     Status      Name      Device  Type  BlkSz  Size      Blocks
==============================================================================
0.0.0100   active      dasda     94:0    ECKD  4096   7042MB    1802880
0.0.0301   active      dasdb     94:4    FBA   512    512MB     1048576
0.0.0300   active      dasdc     94:8    FBA   512    256MB     524288
0.0.0101   active      dasdd     94:12   ECKD  4096   6676MB    1709280
0.0.2100   n/f         dasde     94:16   ECKD
0.0.2101   n/f         dasdh     94:28   ECKD
```

5. In this example, the target minidisks (2100-2101) are named /dev/dasdh and /dev/dasde. Format the target devices with the **dasdfmt** command using a 4096 byte (4 KB) block size:

```
# dasdfmt -b 4096 -y -f /dev/dasde
Finished formatting the device.
Rereading the partition table... ok
# dasdfmt -b 4096 -y -f /dev/dasdh
Finished formatting the device.
Rereading the partition table... ok
```

6. Disable the devices with the chccwdev -d command:

```
# chccwdev -d 2100-2101
Setting device 0.0.2100 offline
Done
Setting device 0.0.2101 offline
Done
```

7. Detach the source disks with the **vmcp DETACH** command:

```
# vmcp det 2100-2101
2100-2101 DETACHED
```

It is now time to configure it for kickstart. LINUX1 must have its own parameter and configuration files, which are again based on the RH64GOLD virtual machine. Perform the following steps:

1. **LOGOFF** of MAINT and logon to LNXMAINT.
2. Copy the parameter and configuration files from RH64GOLD to LINUX1 as follows:

```
==> copy rh64gold * d linux1 = =
```

3. Edit the LINUX1 PARM-RH6 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 “kick starts”, and the **cmdline** option prevents the installer’s text-based user interface from opening on the 3270 console:

```
==> x linux1 parm-rh6 d
ramdisk_size=40000 root=/dev/ram0 ro ip=off
CMSDASD=191 CMSCONFFILE=linux1.conf-rh6
ks=nfs:9.12.7.7:/var/nfs/ks/linux1-ks.cfg
```

```
RUNKS=1 cmdline
```

4. Next, edit the `LINUX1 CONF` file, and change the DASD range and networking information:

```
==> x linux1 conf-rh6 d
DASD=100-101,300-301
HOSTNAME=virtcook1.itso.ibm.com
NETTYPE=qeth
IPADDR=9.12.7.1
...
====> file
```

5. Logoff of `LNXMAINT`.

You should now be ready to *kickstart* RHEL 6.4 onto a virtual machine.

12.3 Kickstart the LINUX1 user

Perform the following steps to kickstart the `LINUX1` user:

1. Logon to `LINUX1`. When asked to IPL from disk `100`, answer `n`:

```
LOGON LINUX1
...
Do you want to IPL Linux from minidisk 100? y/n
n
```

2. Add more memory for the installation process. Temporarily modify the storage up to 1 GB with the **DEFINE STORAGE** command. Then, **IPL CMS** and again answer **n** to the question of IPLing Linux:

```
==> def stor 1g
00: STORAGE = 1G
00: Storage cleared - system reset.
==> ipl cms
...
Do you want to IPL Linux from minidisk 100? y/n
n
```

Verify that you have a 1 GB virtual machine:

```
==> q v stor
00: STORAGE = 1G
```

This change is for the duration of the virtual machine session. When you log off and log back on this virtual machine, the storage will go back to 256 MB.

3. Run **RHEL64 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:** Toward the end of the kickstart, it is normal to see some unrecognized characters on the panel. This is because the 3270 console cannot display the progress meter during the postinstallation phase. To automatically clear the 3270 console and avoid multiple screens of unreadable messages, issue the **#cp term more 0 0** command before running **RHEL64 EXEC**.

```
==> rhel64
...
Kernel command line: ramdisk_size=40000 root=/dev/ram0 ro ip=off
                     CMSDASD=191 CMSCONFFILE=linux1.conf-rh6
                     ks=nfs:9.12.7.7:/var/nfs/ks/linux1-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 installations, 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...
```

Congratulations. You have now installed Red Hat Enterprise Linux onto the virtual server using kickstart. This process can be repeated in the future for other Linux guests. This chapter has shown a minimal installation with kickstart. However, you can completely customize the kickstart file to install different packages based on your requirements. For more information about kickstart options, see the documentation at the following website:

https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Installation_Guide/s1-kickstart2-options.html

From there, click **Installation Guide**, then **28. Kickstart Installations**.

You might now want to configure this image as the golden image was. If so, see section 9.2, “Configure the golden image” on page 183.

Postinstallation section of the kickstart file

You could also add all extra installation processes to the `%post` section of the kickstart file. This section is very flexible and can do most of the postinstallation process, such as file editing, service configuration, package installation, and much more. For example, to set up the automount service and to setup yum on the post section, follow these instructions:

```
%post --interpreter=/bin/bash

#Set up automount for this installation of RHEL
cat >> /etc/auto.master <<\EOF
/var/nfs    /etc/auto.nfs
EOF
cat >> /etc/auto.nfs <<\EOF
rhel64     -ro,hard,intr 9.12.7.7:/var/nfs/rhel64
EOF
service autofs reload

#Set up yum to access internal RHEL package repository
cat > /etc/yum.repo.d/rhel64.repo <<\EOF
[RHEL64]
name=Red Hat Enterprise Linux 6.4
baseurl=file:///var/nfs/rhel64/Server
EOF
rpm --import /var/nfs/rhel64/RPM-GPG-KEY-redhat-release
```

Nearly any configuration can be scripted and used in the %postinstallation section of the kickstart file.

13 Service Linux with the Red Hat Network

"*The faster you go, the shorter you are.*"

— Albert Einstein

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://access.redhat.com

The following sections describe how to configure a Linux guest for **yum**, and manage the guest through RHN:

- "Register your system with RHN" on page 235
- "Installing and updating packages using yum" on page 237
- "Manage your systems through the RHN" on page 238

13.1 Register your system with RHN

This section assumes that you have already obtained a valid entitlement for Red Hat Enterprise Linux for System z, or have completed the steps to obtain an evaluation copy. To receive a 90-day evaluation at no charge, visit the following website:

http://www.redhat.com/z

Note: To download a Red Hat Evaluation, ensure that you already have a Red Hat Network Account or create a new one.

Select the **Evaluation Software Download** link on the left, under Overview. Follow the instructions on the next window to download the evaluation. After completing the form, you will soon receive an email with activation instructions.

13.2 Register your system with Red Hat Network

> **Note:** Red Hat Network offers a new registration, **Certificate-based subscription management**, which allows users to easily track their subscription quantity and usage. To learn more about it, see the following site:
>
> `https://access.redhat.com/knowledge/node/63568`
>
> To learn more about the differences about the Certificate-based RHN and the RHN Classic, see the following site:
>
> `https://access.redhat.com/knowledge/articles/63269`

From the command line, perform the following steps:

1. Start an SSH session as `root` to the system that you want to register.
2. Start the registration process:

   ```
   # rhn_register
   ```

   The Red Hat Network Text User Interface appears, as shown below.

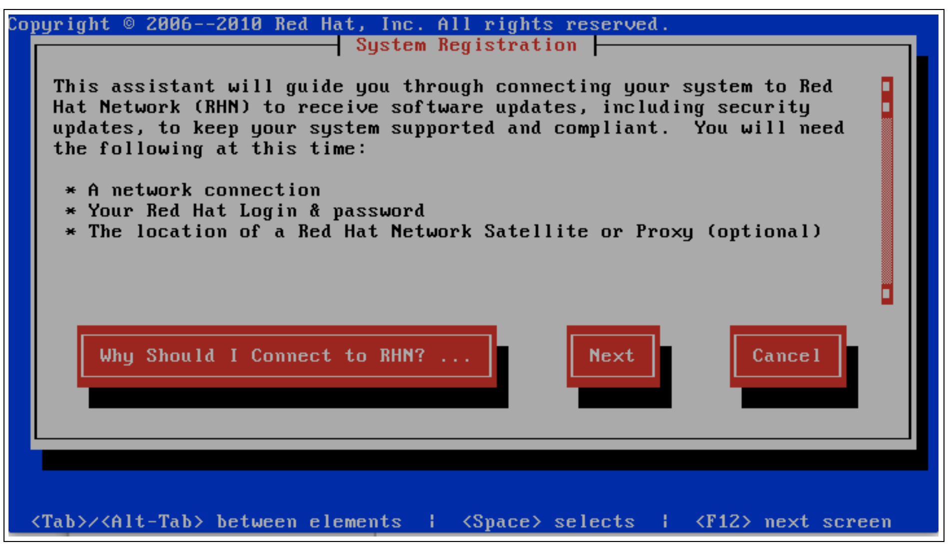


Figure 13-1 Red Hat Network User Interface panel

3. Select **Next** and enter your Red Hat Login and Password. Select **Next** again.
4. Select the option **All Available Updates** then **Next**.
5. Select **OK**.
6. Type the name of the profile for this system on Red Hat Network (RHN).
7. Accept the default package list unless you do not want to receive updates for a selected number of packages, then click **Next**.
8. Select **Next** to send the profile, then click **OK** to complete the registration of your system.

Now your system is ready to access Red Hat and install or update packages.

To fully update your system, invoke the **yum update -y** command.

13.3 Installing and updating packages using yum

> **Note:** The systems installed during this book will use an internal package repository to perform package installations. For production systems, use the RHN Satellite (or Red Hat Network Satellite) for a complete lifecycle management solution.

You can choose to perform these steps first on a "clone", such as LINUX1, then later on the golden image. In this fashion, you can test the process on an appliance that can be discarded, and later when all is tested and working, update the golden image so that all clones created thereafter are enabled for RHN.

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 /etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
# rhnreg_ks --username=myuser --password=mypw --profilename=virtcook1.itso.ibm.com
```

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-4.4.1
# yum upgrade cpp
Loading "rhnplugin" plugin
Loading "installonlyn" plugin
Setting up Upgrade Process
Setting up repositories
rhel-s390x-server-6 100% |=========================|  950 B    00:00
...
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
  Updating  : cpp                          ######################### [1/2]
  Cleanup   : cpp                          ######################### [2/2]

Updated: cpp.s390x Complete!
```

Now query the cpp package and you should see that it has been updated:

```
# rpm -q cpp
cpp-4.4.6-4.el6.s390x
```

To update every installed package on the system, run the following command:

```
# yum upgrade
```

For more information about the **yum** command, see the yum(8) man page.

13.4 Manage your systems through the RHN

You can also manage the packages on this Linux guest through the web interface at the following customer portal:

http://access.redhat.com

Click Subscriptions, log in to Red Hat Network, then RHN Classic.

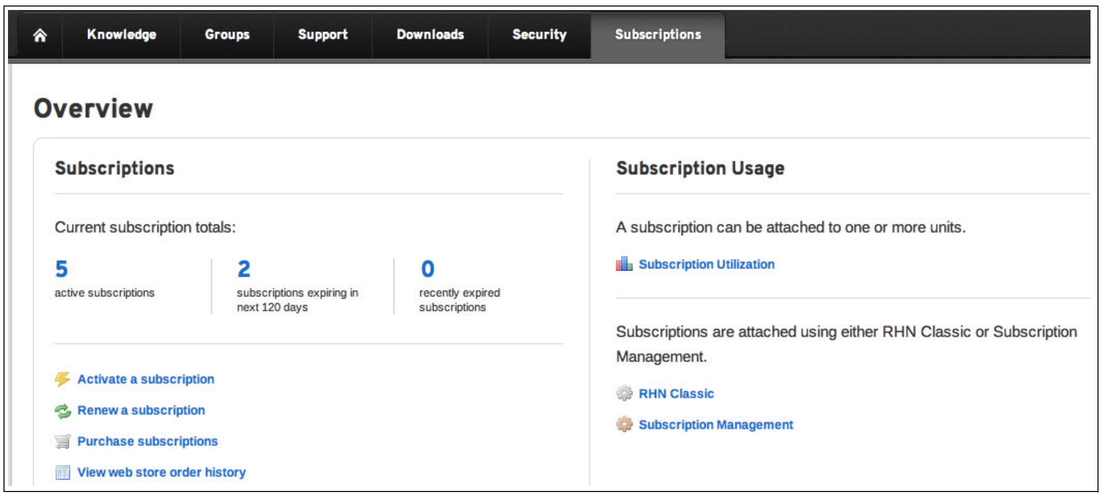


Figure 13-2 Example of the Red Hat Network Classic view

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, the system is fully updated.

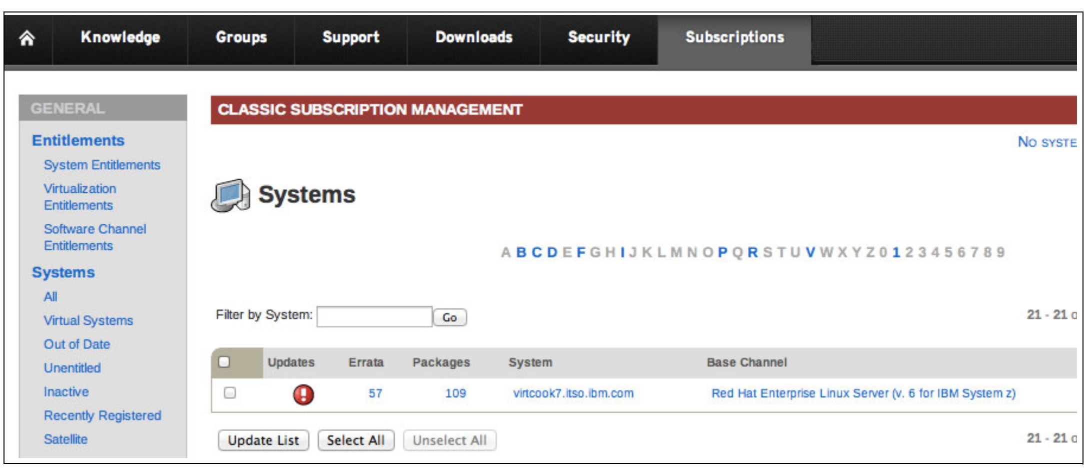


Figure 13-3 RHN Classic Systems overview

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 13-4 on page 239 shows. Click the **Software** tab to view all packages that are installed on this system. From this tab, you can also update, remove, or install new packages onto the system.

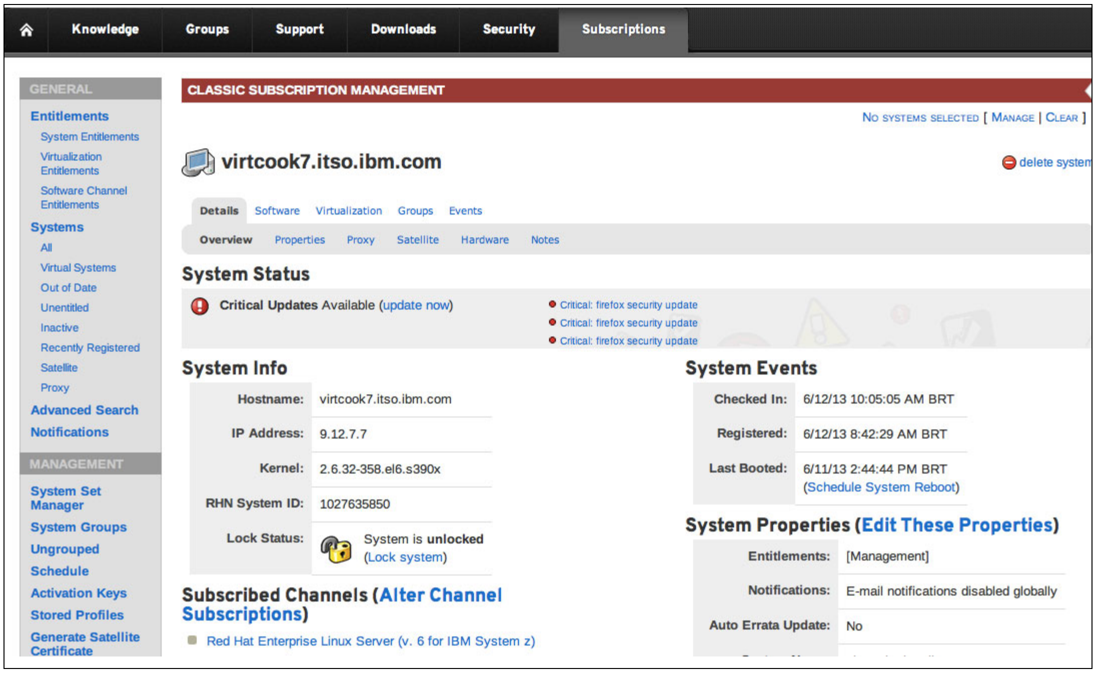


Figure 13-4 RHN Classic System details

For more information about managing your systems through RHN, including usage guides and frequently asked questions, see the following customer portal:

http://access.redhat.com/site/documentation

14

Red Hat Network Satellite Server

> "*Insanity: doing the same thing over and over again and expecting different results.*"
>
> — Albert Einstein

This chapter describes Red Hat Network Satellite Server, which is a systems management platform for your Linux infrastructure. It is built on open standards and uses a web-based graphical interface. Its services are provided through functional modules that allow you to enhance management capabilities for Red Hat Enterprise Linux. For more information about Red Hat Network Satellite Server, see the following website:

http://www.redhat.com/satellite

Note: To install the Red Hat Network Satellite, you need a valid RHN Satellite subscription and a valid RHN Satellite certificate (issued by Red Hat as part of the RHN Satellite subscription).

14.1 Prepare to install the RHN Satellite

To prepare to install the RHN Satellite Server, perform the following steps:

1. 14.1.1, “Add disks to LNXADMIN” on page 242
2. 14.1.2, “Utilize the new disk space” on page 242
3. 14.1.3, “Extend existing logical volumes” on page 244

14.1.1 Add disks to LNXADMIN

The RHN Satellite Server will be installed on the `LNXADM` virtual machine. To accomplish this, approximately 35 GB of disk space will be added. To do so, perform the following steps:

1. Log in as `MAINT`.
2. Format the disks necessary and attach them to `SYSTEM`. In this example, 3390-9 volumes at real device addresses ***1360***, ***1361***, ***1365***, ***1366***, ***136F***, and ***136A*** are added. Details are described in section 5.8.3, “Use the CPFORMAT EXEC” on page 83.
3. Add the disks line. In this example, lines are added at the bottom of the `SUBCONFIG LNXADM-1` as minidisks `200-205`:

```
SUBCONFIG LNXADM-1
   MDISK 0100 3390 0001 10016 JM1263 MR LNX4VM LNX4VM LNX4VM
   MDISK 0101 3390 0521 9496 JM1262 MR LNX4VM LNX4VM LNX4VM
   MDISK 0200 3390 0001 10016 JM1360 MR LNX4VM LNX4VM LNX4VM
   MDISK 0201 3390 0001 10016 JM1361 MR LNX4VM LNX4VM LNX4VM
   MDISK 0202 3390 0001 10016 JM1365 MR LNX4VM LNX4VM LNX4VM
   MDISK 0203 3390 0001 10016 JM1366 MR LNX4VM LNX4VM LNX4VM
   MDISK 0204 3390 0001 10016 JM136f MR LNX4VM LNX4VM LNX4VM
   MDISK 0205 3390 0001 10016 JM136A MR LNX4VM LNX4VM LNX4VM
```

4. Log on to `LNXADMIN`, shut down Linux, and log off.
5. Log back on to `LNXADMIN` and start Linux.

The virtual machine should now have minidisks at virtual addresses `200-205`.

14.1.2 Utilize the new disk space

To use the new disk space, perform the following steps:

1. Open a Secure Shell (SSH) connection to `LNXADM` and use the following commands to link the new minidisks. In this example, they are at `200 - 205`:

```
# vmcp link lnxadm-1 200 200 mr
# vmcp link lnxadm-1 201 201 mr
# vmcp link lnxadm-1 202 202 mr
# vmcp link lnxadm-1 203 203 mr
# vmcp link lnxadm-1 204 204 mr
# vmcp link lnxadm-1 205 205 mr
```

2. To make the minidisks available to Linux, enable them with the **`chccwdev -e`** command:

```
# chccwdev -e 200
# chccwdev -e 201
# chccwdev -e 202
# chccwdev -e 203
# chccwdev -e 204
# chccwdev -e 205
```

3. Use the **lsdasd** command to list the disks that are available to LNXADMIN:

```
# lsdasd
Bus-ID     Status      Name      Device  Type  BlkSz  Size      Blocks
==============================================================================
0.0.0100   active      dasda     94:0    ECKD  4096   7042MB    1802880
0.0.0301   active      dasdb     94:4    FBA   512    512MB     1048576
0.0.0300   active      dasdc     94:8    FBA   512    256MB     524288
0.0.0101   active      dasdd     94:12   ECKD  4096   6676MB    1709280
0.0.0200     n/f       dasde     94:16   ECKD  4096   7042MB    1802880
0.0.0203     n/f       dasdf     94:20   ECKD  4096   7042MB    1802880
0.0.0202     n/f       dasdg     94:24   ECKD  4096   7042MB    1802880
0.0.0204     n/f       dasdh     94:28   ECKD  4096   7042MB    1802880
0.0.0201     n/f       dasdi     94:32   ECKD  4096   7042MB    1802880
0.0.0205     n/f       dasdj     94:36   ECKD  4096   7042MB    1802880
```

4. Format the minidisks for Linux with the **dasdfmt** command. In this example, disks dasde - dasdj are formatted:

```
for i in e f g h i j
do
  dasdfmt -b 4096 -y -f /dev/dasd$i &
done
```

5. Add the minidisks to the /etc/dasd.conf configuration file:

```
# vi /etc/dasd.conf
0.0.0301 use_diag=0 readonly=0 erplog=0 failfast=0
0.0.0300 use_diag=0 readonly=0 erplog=0 failfast=0
0.0.0101 use_diag=0 readonly=0 erplog=0 failfast=0
0.0.0100 use_diag=0 readonly=0 erplog=0 failfast=0
0.0.0200 use_diag=0 readonly=0 erplog=0 failfast=0
0.0.0201 use_diag=0 readonly=0 erplog=0 failfast=0
0.0.0202 use_diag=0 readonly=0 erplog=0 failfast=0
0.0.0203 use_diag=0 readonly=0 erplog=0 failfast=0
0.0.0204 use_diag=0 readonly=0 erplog=0 failfast=0
0.0.0205 use_diag=0 readonly=0 erplog=0 failfast=0
```

   These minidisks will now be enabled at boot time.

6. Create a single partition of each minidisk with the **fdasd -a** command:

```
# fdasd -a /dev/dasde
# fdasd -a /dev/dasdf
# fdasd -a /dev/dasdg
# fdasd -a /dev/dasdh
# fdasd -a /dev/dasdi
# fdasd -a /dev/dasdj
```

7. Create a physical volume from each partition for LVM:

```
# pvcreate /dev/dasde1
  Physical volume "/dev/dasde1" successfully created
# pvcreate /dev/dasdf1
  Physical volume "/dev/dasdf1" successfully created
# pvcreate /dev/dasdg1
  Physical volume "/dev/dasdg1" successfully created
# pvcreate /dev/dasdh1
  Physical volume "/dev/dasdh1" successfully created
# pvcreate /dev/dasdi1
  Physical volume "/dev/dasdh1" successfully created
# pvcreate /dev/dasdj1
  Physical volume "/dev/dasdh1" successfully created
```

Six physical volumes comprising approximately 35 GB of new disk space should now be available for the Linux system to use.

14.1.3 Extend existing logical volumes

Logical volumes can now be made larger. To do so, perform the following steps:

1. Extend the LVM volume group that is named `var_vg`. In this example, the disks are `dasde1` - `dasdj1`:

```
# vgextend var_vg /dev/dasde1
  No physical volume label read from /dev/dasde1
  Physical volume "/dev/dasde1" successfully created
  Volume group "var_vg" successfully extended
# vgextend var_vg /dev/dasdf1
  No physical volume label read from /dev/dasdf1
  Physical volume "/dev/dasdf1" successfully created
  Volume group "var_vg" successfully extended
# vgextend var_vg /dev/dasdg1
  No physical volume label read from /dev/dasdg1
  Physical volume "/dev/dasdg1" successfully created
  Volume group "var_vg" successfully extended
# vgextend var_vg /dev/dasdh1
  No physical volume label read from /dev/dasdh1
  Physical volume "/dev/dasdh1" successfully created
  Volume group "var_vg" successfully extended
# vgextend var_vg /dev/dasdi1
  No physical volume label read from /dev/dasdi1
  Physical volume "/dev/dasdi1" successfully created
  Volume group "var_vg" successfully extended
# vgextend var_vg /dev/dasdj1
  No physical volume label read from /dev/dasdj1
  Physical volume "/dev/dasdj1" successfully created
  Volume group "var_vg" successfully extended
```

2. Add 20 GB of the space to the logical volume `var_lv` with the **`lvextend`** command:

```
# lvextend -L+20G  /dev/var_vg/var_lv
  Extending logical volume var_lv to 30.95 GiB
  Logical volume var_lv successfully resized
```

3. Resize the current file system with the **`resize2fs`** command:

```
# resize2fs /dev/var_vg/var_lv
resize2fs 1.41.12 (17-May-2010)
Filesystem at /dev/var_vg/var_lv is mounted on /var; on-line resizing required
...
```

   This operation does not require the file system to be brought offline.

4. Create a new 4 GB logical volume named `usr_lv` to migrate the contents of the current `/usr/` directory:

```
# lvcreate -L4G -n usr_lv var_vg
  Logical volume "usr_lv" created
```

5. Create an ext4 file system of the new logical volume `usr_lv`:

```
# mkfs.ext4 /dev/var_vg/usr_lv
mke2fs 1.41.12 (17-May-2010)
...
```

The logical volume over `/var/` has now been extended and a new logical volume has been created for `/usr/`.

Move the /usr/ file system off the root file system

The root file system will need more space for the RHN Satellite. To fix this, the contents of /usr/ will be moved to the new logical volume. The system already has a /usr/ directory, so we will migrate the /usr/ contents to the new usr_lv file system and delete the old /usr/. In order to accomplish this task, perform the following steps:

1. Mount the new logical volume over a new directory /usr2/:

   ```
   # mkdir /usr2
   # mount /dev/var_vg/usr_lv /usr2
   ```

2. Copy the contents of the /usr/ to /usr2/ with the **cp -a** command:

   ```
   # cp -a /usr/* /usr2/
   ```

   Recursively delete the contents of /usr/ to free up space in the root file system:

   ```
   # rm -rf /usr
   ```

3. Unmount the newly populated logical volume and mount it over /usr/:

   ```
   # umount /usr2
   # mount /dev/var_vg/usr_lv /usr
   ```

4. Reset the labels of all the files in /usr/ for SE Linux:

   ```
   # restorecon -Rv /usr
   ```

The /usr/ file system has now been moved off the root file system.

The next step is to create a new logical volume for the RHN Satellite's database:

1. Create a new logical volume named rhn_sat:

   ```
   # lvcreate -L15G -n rhn_sat var_vg
     Logical volume "rhn_sat" created
   ```

2. Create an ext4 file system of the new logical volume:

   ```
   # mkfs.ext4 /dev/var_vg/rhn_sat
   mke2fs 1.41.12 (17-May-2010)
   ...
   ```

A logical volume of 15 GB has now been created for Red Hat Satellite.

Start the RHN Satellite Server

To start the RHN Satellite Server installation, perform the following steps:

> **Note:** In this example, before starting the RHN Satellite Server installation, increase the VM memory to 2 GB.

1. Copy the Red Hat Network Satellite Server ISO image to /root/.
2. Copy the RHN Satellite Digital Certificate to /root/.
3. Create a mount point directory for the ISO image:

   ```
   # mkdir /root/sat
   ```

   a. Mount the RHN Satellite ISO image:

   ```
   # mount -o loop /root/satellite5.5*s390x.iso /root/sat
   ```

4. Start the RHN Satellite installation process.

Note: Disconnected installations are for systems that do not have access to the Internet. To perform a disconnected installation, use the following command:

```
# ./install --disconnected
```

This installation used the connected mode, where the system has access to the Internet.

```
# cd /root/sat
# ./install.pl
Starting the Red Hat Network Satellite installer.
Performing preinstall checks.
Preinstall checks complete. Beginning installation.
RHN Registration.
...
Activating RHN Satellite.
Where is your satellite certificate file? /root/satellite.cert
Loading RHN Satellite Certificate.
...
Configuring apache SSL virtual host.
Should setup configure apache's default ssl server for you (saves original ssl.conf)
[Y]? Y
/etc/httpd/conf.d/ssl.conf has been backed up to ssl.confswsave
...
Creating SSL certificates.
CA certificate password? <type your password here)>
Reenter CA certificate password?
Organization? itso
Organization Unit [virtcook7.itso.ibm.com]?
Email Address [root@localhost]?
City? Poughkeepsie
State? NY
Country code (Examples: "US", "JP", "IN", or type "?" to see a list)? US
** SSL: Generating CA certificate.
** SSL: Deploying CA certificate.
** SSL: Generating server certificate.
** SSL: Storing SSL certificates.
...
Setting up Cobbler..
cobblerd does not appear to be running/accessible
Cobbler requires tftp and xinetd services be turned on for PXE provisioning
functionality. Enable these services [Y]? Y
...
Installation complete.
```

5. Visit `https://virtcook7.itso.ibm.com` to create the RHN Satellite administrator account.

   Open a browser to the URL ***https://virtcook7.itso.ibm.com***. You will be redirected to the RHN Satellite administrator creation panel. See Figure 14-1 on page 247.

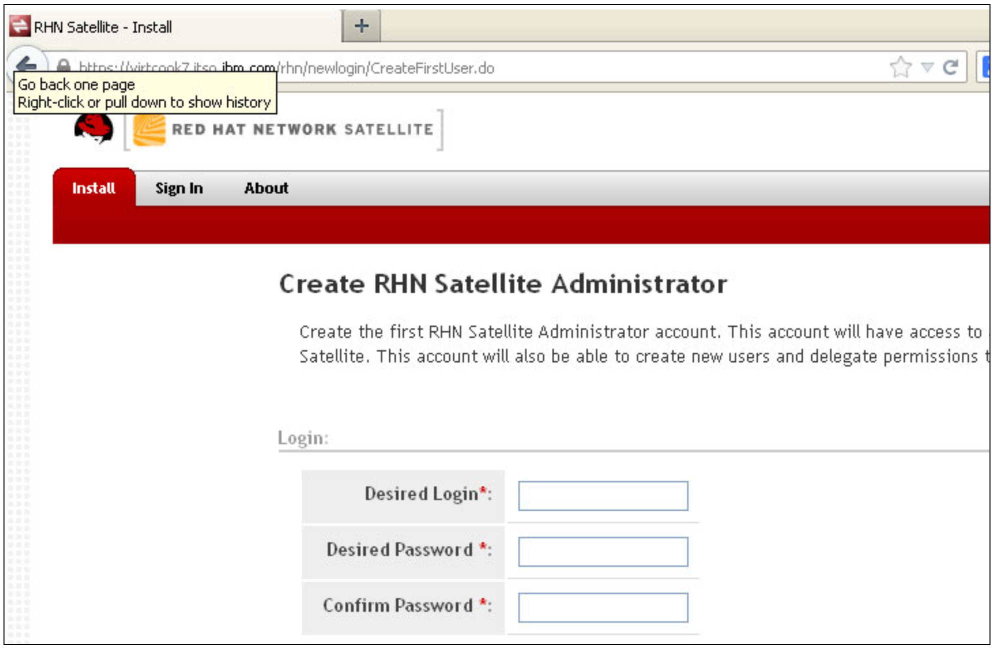


Figure 14-1 Create RHN Satellite Administrator account panel

6. Now access *https://virtcook7.itso.ibm.com* to have access to the login panel. Enter the user and password that you just created to access the system overview dashboards from the RHN Satellite.

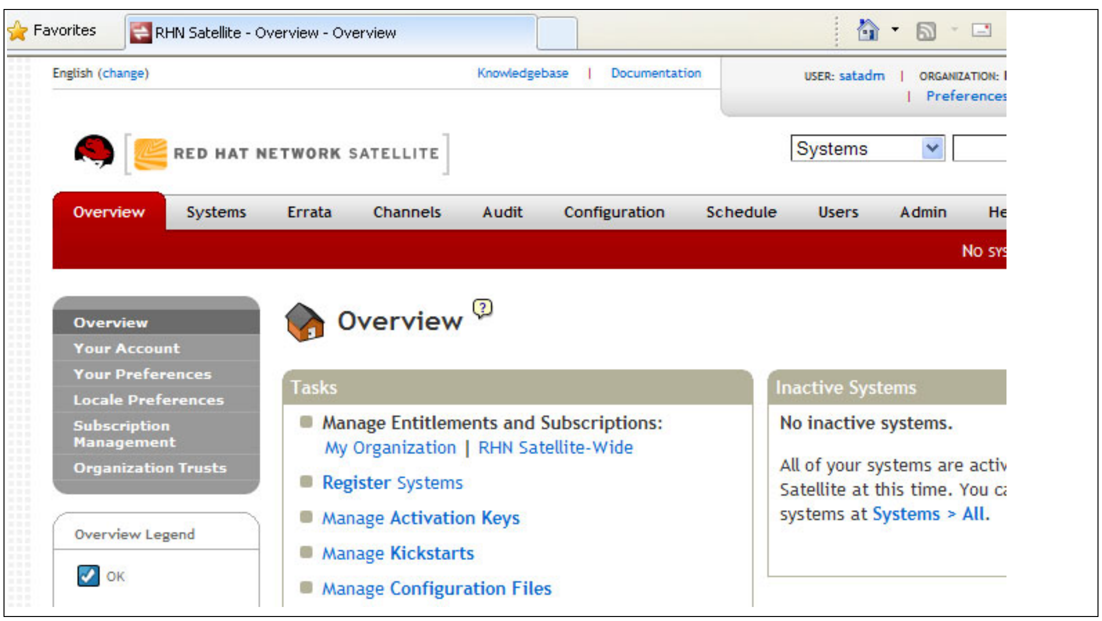


Figure 14-2 RHN Satellite Overview dashboard view

Congratulations. RHN Satellite Server is installed.

To populate RHEL channels in the RHN Satellite's database, perform the following steps:

1. The RHN Satellite Server can handle all software channels for all supported architectures, such as RHEL6 for s390x, RHEL6 for x86_64, RHEL5 for x86_64, RHEL5 for s390x, and more.

In this example, we populate the RHN Satellite's database with RHEL6 for s390x and RHEL6 tools for s390x:

> **Note:** To populate the RHN Satellite' s database using a disconnected mode, download the RHN Satellite Channel ISO images from RHN and consult the RHN Satellite manual at the site that corresponds to your version level:
>
> - `https://access.redhat.com/site/documentation/en-US/Red_Hat_Satellite` (Version 5.6 and later)
> - `https://access.redhat.com/site/documentation/en-US/Red_Hat_Network_Satellite` (Version 5.5 and earlier)

```
# satellite-sync -c rhel-s390x-server-6 -c rhn-tools-rhel-s390x-server-6
07:51:22 Red Hat Network Satellite - live synchronization
07:51:22    url: https://satellite.rhn.redhat.com
07:51:22    debug/output level: 1
07:51:22    db:  rhnsat/<password>@//localhost:1521/rhnsat.world
...
07:51:33 Retrieving / parsing additional arches data
07:51:34 additional arches data complete
07:51:34
07:51:34 Retrieving / parsing channel data
07:52:44    p = previously imported/synced channel
07:52:44    . = channel not yet imported/synced
07:52:44    base-channels:
07:52:44       p rhel-s390x-server-6                      9166
07:52:44    rhel-s390x-server-6:
07:52:44       p rhn-tools-rhel-s390x-server-6              86
...
```

2. The selected channels populate the RHN Satellite' s database. To check the channel contents, open your browser and enter the RHN Satellite URL, then go to Channels. See Figure 14-3.

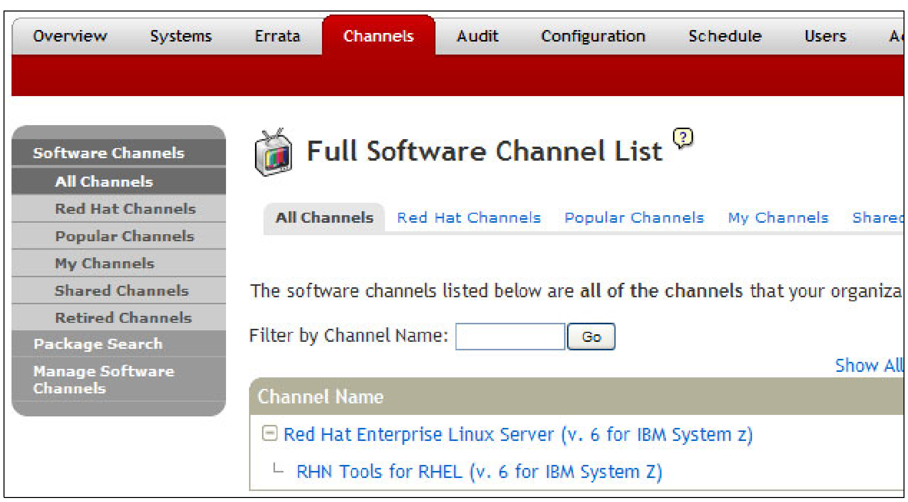


Figure 14-3 RHN Satellite Server Channel List view

Before registering your new systems to the RHN Satellite, set up an activation key that will be used to register new systems to the RHN Satellite by performing the following steps:

1. Enter the **Systems** tab and select **Activation Keys**, as shown in the left screen capture of Figure 14-4 on page 249. Click **Create New Key**. Complete the Description and Key fields,

and select the **Universal Default** option, as shown in the right screen capture of Figure 14-4. As an example, use `rhn_key` for the Description field, and `1-key` for the Key field.

> **Note:** The name of the key in this example is 1-key where the number *one* represents the number of the organization inside the RHN Satellite Server. By using the RHN Satellite, we can manage multiple organizations.

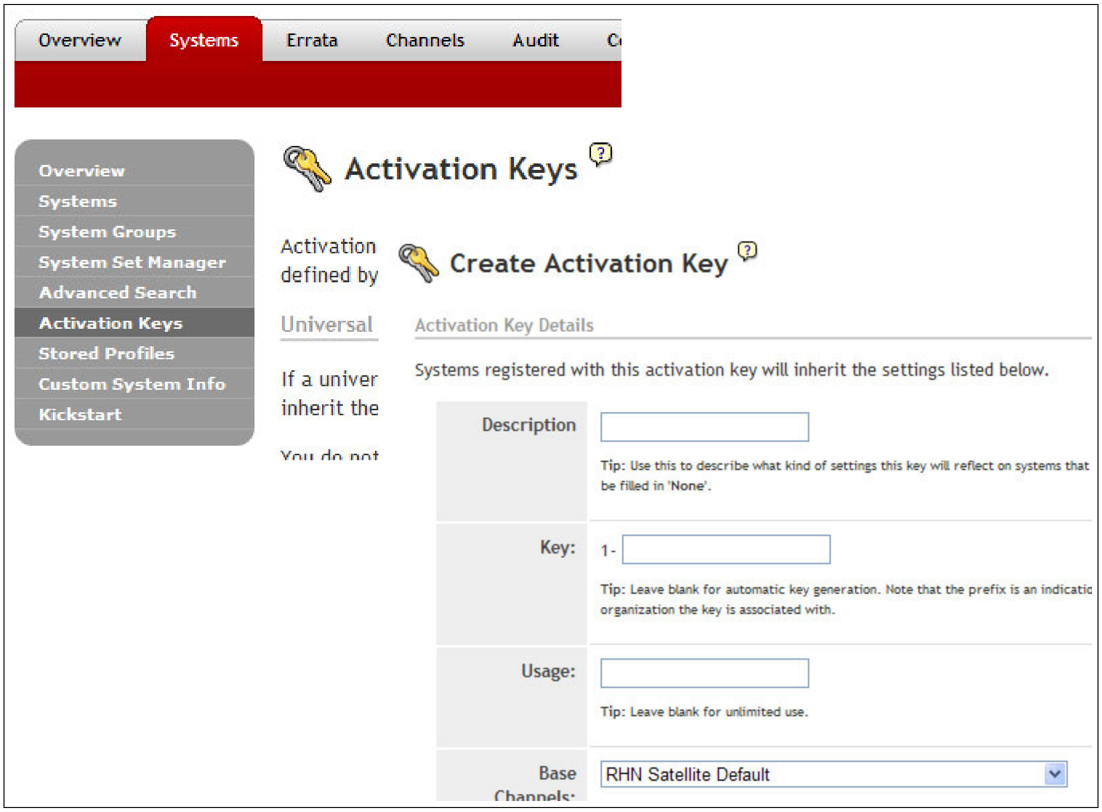


Figure 14-4 RHN Satellite Activation Key view

To register RHEL clients to the RHN Satellite, perform the following commands from an installed Linux guest. In this example, `LINUX2` is used to set up the client side to access the RHN Satellite Server:

```
# wget  https://virtcook7.itso.ibm.com/pub/bootstrap/bootstrap.sh
# chmod +x bootstrap.sh
# ./bootstrap.sh
```

Note: The bootstrap script might need to be edited before use.

Do this at the RHN Satellite Server in `/var/www/html/pub/bootstrap.sh`:

```
#echo
#echo
#echo "MINOR MANUAL EDITING OF THIS FILE MAY BE REQUIRED!"
#echo
...
#echo "Enable this script: comment (with #'s) this block (or, at least just"
#echo "the exit below)"
#echo
#exit 1
...
ACTIVATION_KEYS=1-key
...
```

```
RHN Satellite Server Client bootstrap script v4.0

UPDATING RHN_REGISTER/UP2DATE CONFIGURATION FILES
-------------------------------------------------
* downloading necessary files
  client_config_update.py...
  client-config-overrides.txt...
* running the update scripts
  . up2date config file
...
REGISTRATION
------------
* registering
...
*** this system should now be registered, please verify ***

Check the registered system on Satellite's System' s tab
```

- To apply updates to the registered system, click the red Exclamation Mark and select the errata to update. In this example, just one erratum will be applied. See Figure 14-5.

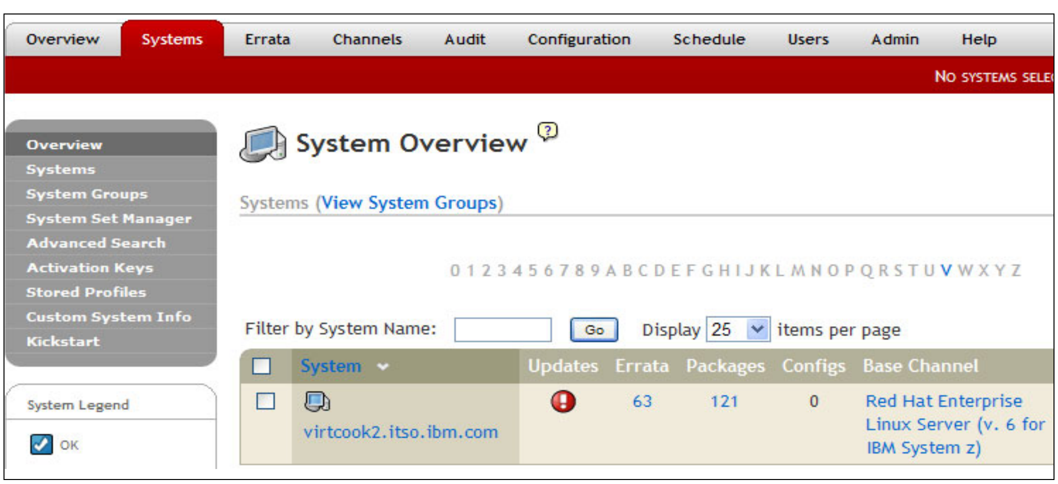


Figure 14-5 RHEL client registered RHN Satellite Server system view

Figure 14-6 on page 251 shows the RHN Satellite view for applying errata.

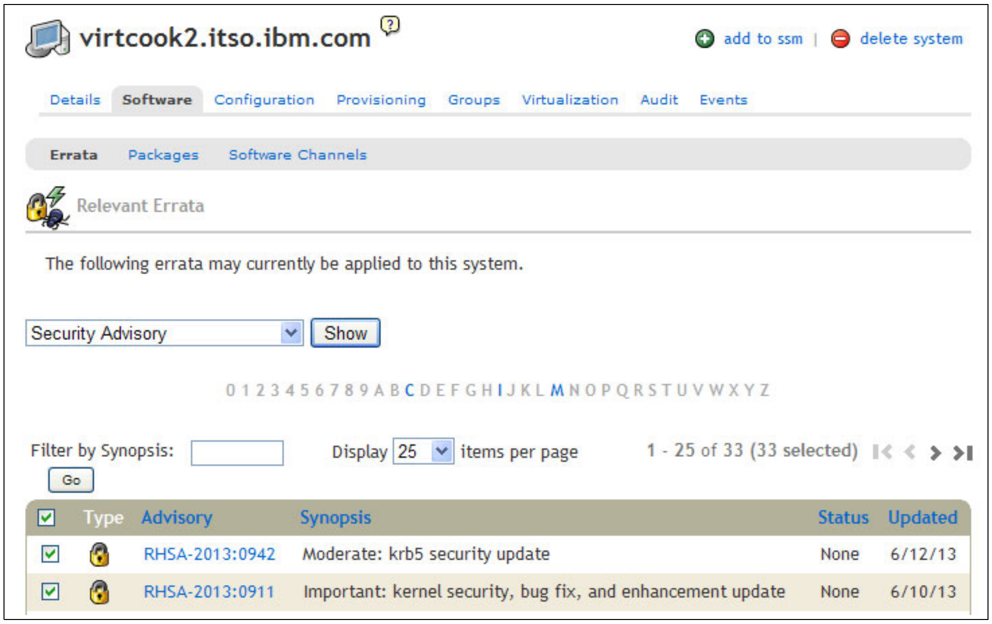


Figure 14-6 RHN Satellite view for applying errata

- Select **Apply Errata**.
- Select when this errata will be applied and confirm.

Note: There is a daemon on each RHEL installation called *rhnsd*. This daemon controls when this client will look for pending actions from either RHN or RHN Satellite. The default period is `30 minutes`.

To push updates directly from the RHN Satellite to the RHEL clients without waiting for the rhnds daemon query for pending actions, refer to the RHN Satellite documentation.

Figure 14-7 shows the RHN Satellite view for scheduling updates.

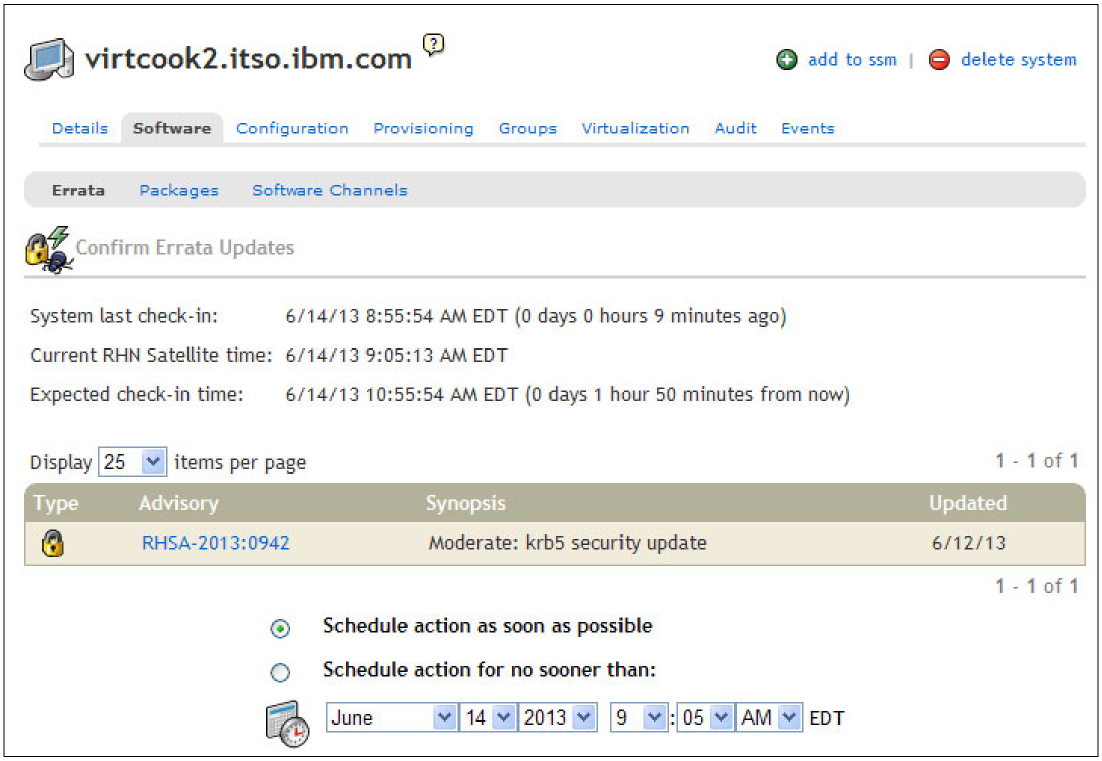


Figure 14-7 RHN Satellite view for scheduling updates

- To get a list of all pending actions, click the **Schedule** tab.

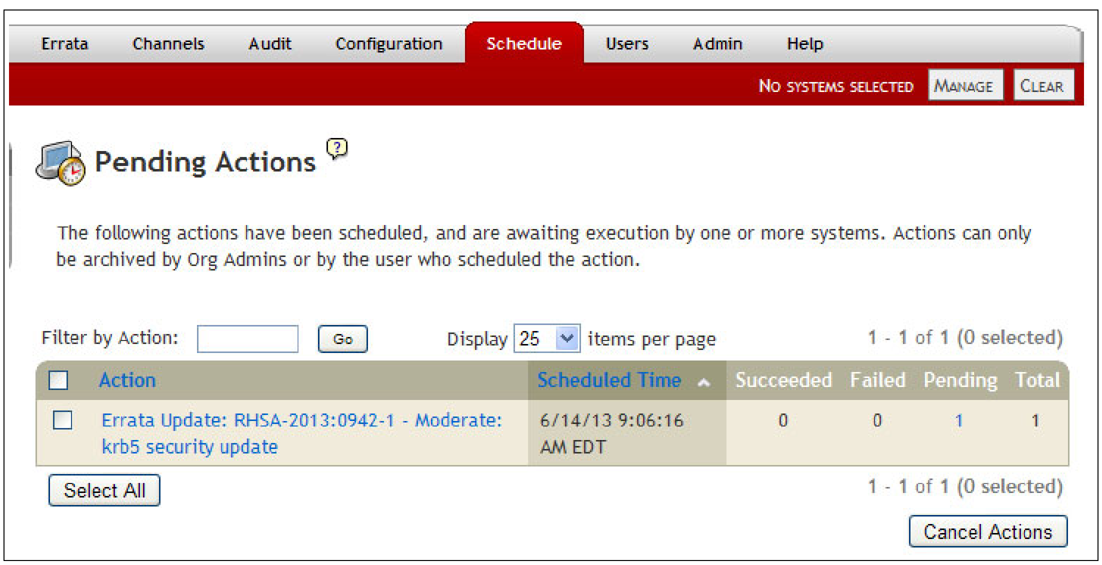


Figure 14-8 RHN Satellite pending actions view

To kickstart a Linux guest, perform the following steps:

1. To create a basic kickstart file, select the **Systems** tab, select **kickstart**, and then click **Create a new kickstart profile**. See Figure 14-9 on page 253.

Note: There is also an option to upload a previously created kickstart file to the RHN Satellite's database. If this is wanted, use the option to upload a new kickstart file.

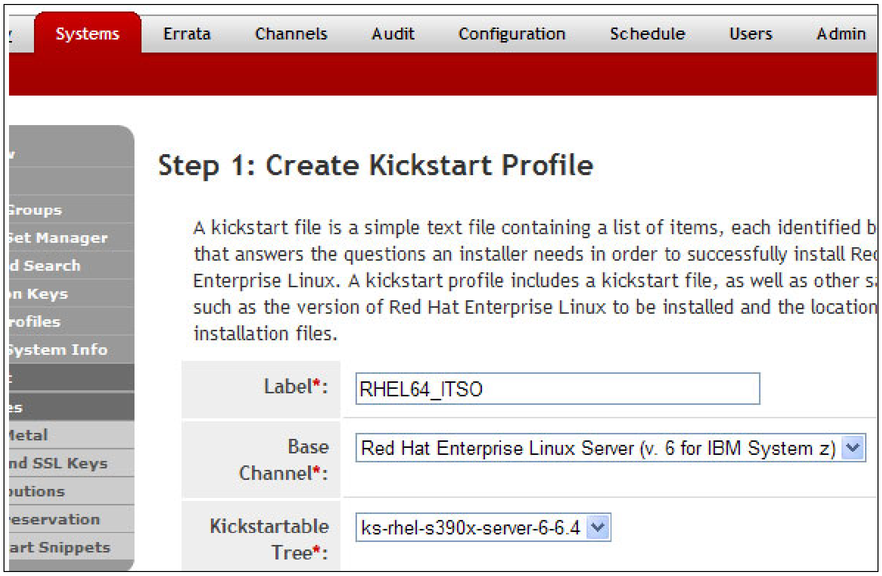


Figure 14-9 RHN Satellite Server view to create a kickstart file

2. Click **Next** and accept the defaults for Step 2.
3. Set up the password for the root user.
4. Keep all options defaults, select **Activation Keys**, and select the key created previously. See Figure 14-10.

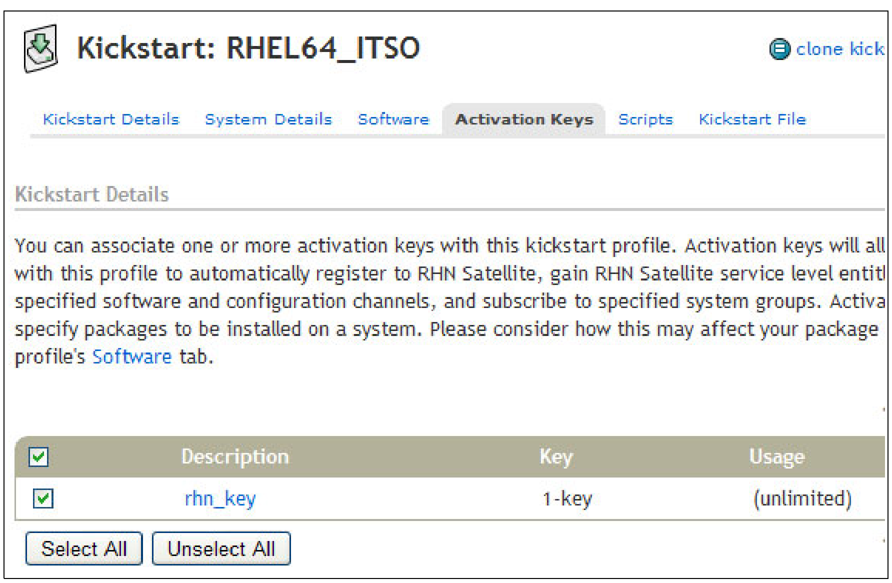


Figure 14-10 RHN Satellite Server view to add an activation key to the kickstart profile

5. Select the kickstart tab to view the complete kickstart file. See Figure 14-11.

Figure 14-11 RHN Satellite Server kickstart profile view

To install a Linux guest using the kickstart created using RHN Satellite Server, perform the following steps:

1. Log in `LNXMAINT` and edit the parm file:

```
==> x linux1 parm-rh6 d
root=/dev/ram0 ro ip=off ramdisk_size=40000
CMSDASD=191 CMSCONFFILE=linux1.CONF-RH6
ks=http://9.12.7.7/ks/cfg/org/1/label/RHEL64_ITSO
RUNKS=1
vnc vncpassword=12345678
```

2. Log in `LINUX1` and start the installation process:

```
==> RHEL64
```

After the installation process (which should occur without the need for user input), check the RHN Satellite Server system tab for the newly created system. See Figure 14-12 on page 255.

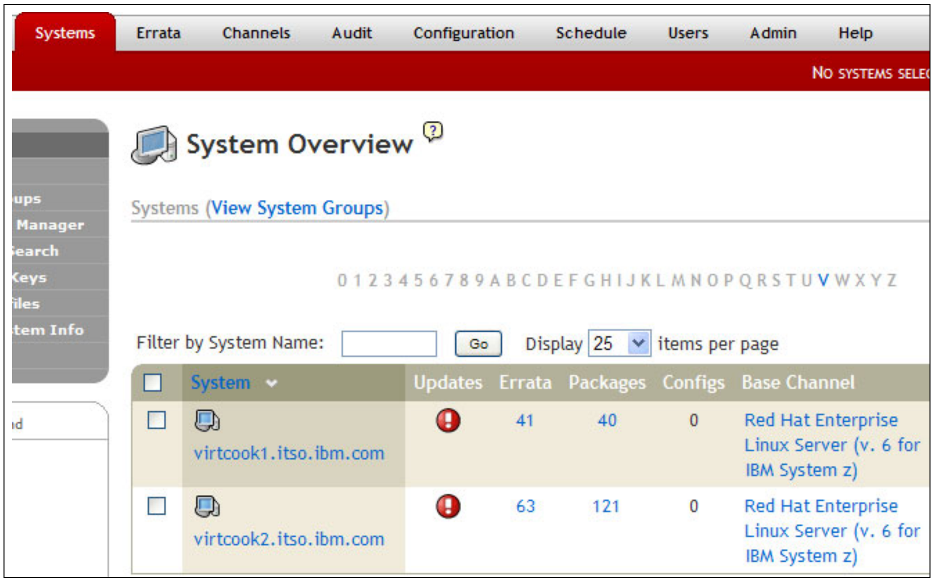


Figure 14-12 RHN Satellite view of a newly created system

To manage the new system, click its name, in this example *virtcook1.itso.ibm.com*, and choose from the list of available options. See Figure 14-13.

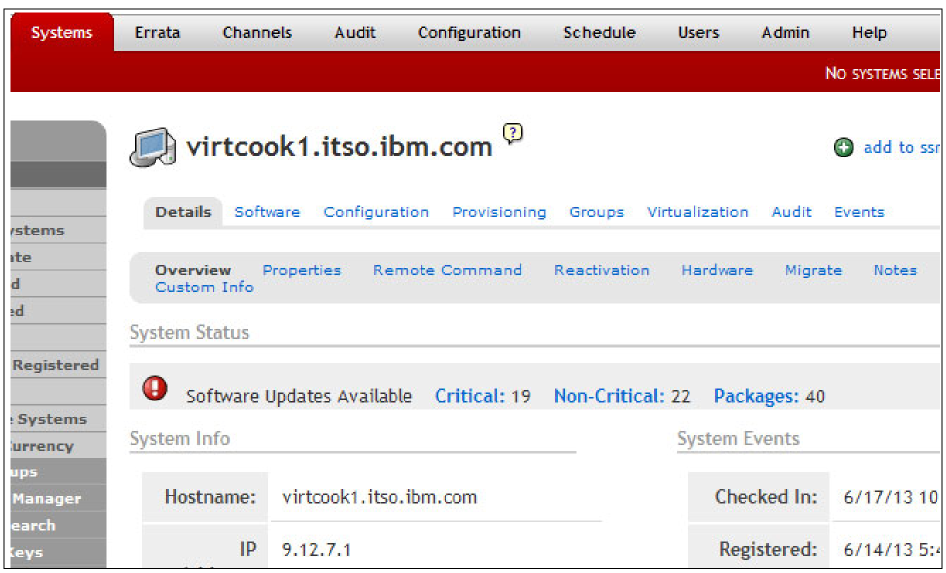


Figure 14-13 RHN Satellite view of system details

Congratulations. You have created a new Linux image using the RHN Satellite.

Part 3

SUSE Linux Enterprise Server 11 SP3

This part of the book focuses on SUSE Linux Enterprise Server. It consists of the following chapters:

- Chapter 15, “Install SUSE Linux Enterprise Server 11 SP3 on LNXADMIN” on page 259; Describes how to install and configure SLES 11 SP3 onto the *Linux administration system*, which does the cloning and other tasks.
- Chapter 16, “Install the SUSE Linux Enterprise Server 11 SP3 golden image” on page 281: Describes how to install and configure two Linux images onto a *golden image*, from which other Linux images will be cloned.
- Chapter 17, “Clone SUSE Linux Enterprise Server 11 SP3” on page 305: Explains how to prepare z/VM virtual machines and clone your first virtual server, both manually and by using a shell script.
- Chapter 18, “Create SUSE Linux Enterprise Server 11 SP3 appliances” on page 315. Describes how to create virtual appliances from cloned SLES 11 SP3 servers.
- Chapter 19, “Installing Linux with AutoYaST2” on page 339: Explains how to use AutoYaST2, which allows you to automatically install Linux using a configuration file.
- Chapter 20, “Creating appliances with KIWI” on page 349. Explains how to create and use appliances and bootable images from configuration files.

15

Install SUSE Linux Enterprise Server 11 SP3 on LNXADMIN

"*The only thing that interferes with my learning is my education.*"

— Albert Einstein

Red Hat Enterprise Linux or SUSE Linux Enterprise Server?: If you are working *only* with RHEL 6.4, you can skip the entire part of this book (six chapters).

If you are working *only* with SUSE Linux Enterprise Server 11 SP3 and you have a single system image (SSI), use this chapter to install on `LNXADMIN` on all members. However, it is only necessary to have the SLES 11 SP3 installation files copied to the Linux system on LNXADMIN on one of the members.

To install and configure SUSE Linux Enterprise Server 11 SP3 onto the `IDENTITY LNXADMIN`, perform the following overall steps:

1. "Review the IDENTITY LNXADMIN" on page 259.
2. "Prepare the SUSE Linux Enterprise Server 11 SP3 bootstrap files" on page 260.
3. "Install SUSE Linux Enterprise Server 11 SP3 onto Linux administration system" on page 262.
4. "Configure the SUSE Linux Enterprise Server 11 SP3 Linux administration system" on page 272.

15.1 Review the IDENTITY LNXADMIN

In this section, you will review the identity that should have been defined in section 5.11, "Create identity LNXADMIN for Linux administration" on page 96.

To accomplish this, perform the following steps:

1. Log on to `MAINT`.

2. Edit the USER DIRECT file:

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

3. Search for the string LNXADMIN and review the definition of the virtual machine:

```
====> /lnxadmin
IDENTITY LNXADMIN LNX4VM 256M 1G BDEG
 INCLUDE LNXDFLT
 BUILD ON ZVM63A USING SUBCONFIG LNXADM-1
 BUILD ON ZVM63B USING SUBCONFIG LNXADM-2
 OPTION LNKNOPAS
SUBCONFIG LNXADM-1
 MDISK 0100 3390 0001 10016 JM1263 MR LNX4VM LNX4VM LNX4VM
 MDISK 0101 3390 0521  9496 JM1262 MR LNX4VM LNX4VM LNX4VM
SUBCONFIG LNXADM-2
 MDISK 0100 3390 0001 10016 JM1362 MR LNX4VM LNX4VM LNX4VM
 MDISK 0101 3390 0001 10016 JM1363 MR LNX4VM LNX4VM LNX4VM
```

SUSE Linux Enterprise Server 11 SP3 will be installed onto LNXADMIN on the second SSI member, ZVM63B in this example.

15.2 Prepare the SUSE Linux Enterprise Server 11 SP3 bootstrap files

To IPL a SUSE Linux Enterprise Server 11 SP3 installation system, the following three bootstrap files must be copied and "punched" to the reader:

- A Linux kernel
- A parameter file
- A Linux initial RAMdisk

Think of these files as a PC Linux boot CD or DVD. A short REXX EXEC is commonly used to clean out the reader, punch the three files and IPL the reader.

To prepare the SUSE Linux Enterprise Server 11 SP3 bootstrap files, perform the following steps:

1. Start an SSH session as **root** to the PC server running NFS.
2. Change directory to the mounted DVD and list the directory boot/s390x/ where the kernel and RAMdisk are located:

```
# cd /srv/nfs/s11s3boot/s390x
# ls -l initrd vmrdr.ikr
-r--r--r--. 1 root root 16285995 Jun  6 10:57 initrd
-r--r--r--. 1 root root  9072128 Jun  6 10:57 vmrdr.ikr
```

3. FTP to the z/VM SSI member where LNXADMIN exists and login as LNXMAINT. In this example, the IP address is ***9.12.7.12***:

```
# ftp 9.12.7.12
Connected to 9.12.7.12.
220-FTPSERVE IBM VM Level 630 at GPOK249.ENDICOTT.IBM.COM, 08:59:44 EST THURSDAY
2009-11-26
220 Connection will close if idle for more than 5 minutes.
Name (9.12.7.11:root): lnxmaint
331 Send password please.
Password: lnx4vm
230 LNXMAINT logged in; working directory = LNXMAINT 191
Remote system type is z/VM.
```

4. Copy the SUSE Linux Enterprise Server 11 SP3 kernel (the file `vmrdr.ikr` copied as `SLES11S3 KERNEL`) and the initial RAMdisk (the file `initrd` copied as `SLES11S3 INITRD`). These files must have a format of fixed 80-byte records and be transferred in binary. This format can be set with the **bin** and **site fix 80** FTP subcommands (if this subcommand fails, try **quote site fix 80**):

```
ftp> cd lnxmaint.192
250 Working directory is LNXMAINT 192
ftp> bin
200 Representation type is IMAGE.
ftp> site fix 80
200 Site command was accepted.
ftp> put vmrdr.ikr SLES11S3.KERNEL
local: vmrdr.ikr remote: SLES11S3.KERNEL
...
ftp> put initrd SLES11S3.INITRD
local: initrd remote: SLES11S3.INITRD
...
ftp> quit
```

5. Log off `MAINT` if you are still logged on.
6. Log on to `LNXMAINT`.
7. Besides the kernel and RAMdisk that you just copied, the file `SLES11S3 EXEC` should exist on the `LNXMAINT 192` disk. Use the **FILELIST** command to verify that the kernel and RAMdisk are copied in fixed-80 byte record format. You should see the following files:

```
==> filel sles11s3 * d
LNXMAINT FILELIST A0  V 169  Trunc=169 Size=3 Line=1 Col=1 Alt=0
Cmd   Filename Filetype Fm Format Lrecl    Records     Blocks   Date     Time
      SLES11S3 INITRD   D1 F         80     203575       3977  7/15/13 13:36:42
      SLES11S3 KERNEL   D1 F         80     113402       1749  7/15/13 13:36:34
      SLES11S3 EXEC     D1 V         63         12          1  6/25/13 10:09:16
```

8. Quit by pressing **F3**.
9. Verify that the **SLES11S3 EXEC** file has the correct information. The kernel and RAMdisk have hardcoded file names, but the file name of the parameter file will be the user ID (**userid()** function) of the user running the EXEC:

```
==> type sles11s3 exec d
/* Punch a SLES 11 SP3 install system to reader and IPL it   */
Address 'COMMAND'
'CP SPOOL PUN *'
'CP CLOSE RDR'
'CP PURGE RDR ALL'
'PUNCH SLES11S3 KERNEL * (NOHEADER'
'PUNCH' 'USERID'() 'PARM-S11 * (NOHEADER'
'PUNCH SLES11S3 INITRD * (NOHEADER'
'CP CHANGE RDR ALL KEEP'
'CP IPL 00C CLEAR'
Exit
```

10. A sample parameter file named `SAMPLE PARM-S11` is provided to save typing. View it with the **TYPE** command:

```
==> type sample parm-s11 d
ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dumb
HostIP=n.n.n.n         Hostname=yourhost.example.com
Gateway=n.n.n.n         Netmask=255.255.255.0 Layer2=1
ReadChannel=0.0.0600  WriteChannel=0.0.0601  DataChannel=0.0.0602
Nameserver=n.n.n.n         portname=whatever portno=0
Install=nfs://n.n.n.n/srv/nfs/s11s3/
```

```
UseSSH=1 SSHPassword=12345678
UseVNC=1  VNCPassword=12345678
InstNetDev=osa OsaInterface=qdio OsaMedium=eth Manual=0
```

11. Copy the sample parameter file to a new file with the file name `LNXADMIN`. Edit the file and update the networking variables with the values correct for your site. Refer to the worksheet in section 2.9.6, “Linux resources worksheet” on page 33. The fields that you should change are in ***bold-italics***. The examples used in this book are as follows:

```
==> copy sample parm-s11 d lnxadmin = =
==> x lnxadmin parm-s11 d
ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dumb
HostIP=9.12.7.8        Hostname=virtcook8.itso.ibm.com
Gateway=9.12.4.1        Netmask=255.255.240.0 Layer2=1
ReadChannel=0.0.0600  WriteChannel=0.0.0601  DataChannel=0.0.0602
Nameserver=9.12.6.7        portname=whatever portno=0
Install=nfs://9.12.5.251/srv/nfs/s11s3
UseSSH=1 SSHPassword=12345678
UseVNC=1  VNCPassword=12345678
InstNetDev=osa OsaInterface=qdio OsaMedium=eth Manual=0
```

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

You are now ready to start the installation.

15.3 Install SUSE Linux Enterprise Server 11 SP3 onto Linux administration system

In this section, you will install Linux onto the `LNXADMIN` virtual machine:

1. Log on to `LNXADMIN`. The common `PROFILE EXEC` should run. You should see a virtual NIC being created at virtual addresses `600 - 602`. You should also see two virtual disks created for swap spaces at virtual addresses `300` and `301`:

```
00: z/VM Version 6 Release 3.0, Service Level 1201 (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 11:38:32 EDT MONDAY 09/02/13
00: Command complete
00: NIC 0600 is created; devices 0600-0602 defined
00: NIC 0600 is connected to VSWITCH SYSTEM VSW1
z/VM V6.3.0    2013-06-24 16:58

DMSACP723I A (191) R/O
DMSACP723I C (592) R/O
DIAG swap disk defined at virtual address 300 (64989 4K pages of swap space)
DIAG swap disk defined at virtual address 301 (129981 4K pages of swap space)
Do you want to IPL Linux from minidisk 100? y/n
```

2. Answer no (**n**) to the question asking you to IPL Linux from `100`:

```
Do you want to IPL Linux from minidisk 100? y/n n
...
```

3. Define the memory size (storage) to 1 GB with the **DEFINE STORAGE** command:

```
==> def stor 1g
00: STORAGE = 1G
00: Storage cleared - system reset.
```

4. IPL CMS, press **Enter** at the `VM READ` prompt and again answer **n**o to the IPL question:

```
==> ipl cms
00: IPL CMSn
z/VM V6.2.0   2011-11-15 11:26

DMSACP723I A (191) R/O
DMSACP113S C(592) not attached or invalid device address
DIAG swap disk defined at virtual address 300 (64989 4K pages of swap space)
DIAG swap disk defined at virtual address 301 (129981 4K pages of swap space)
Do you want to IPL Linux from minidisk 100? y/n
n
```

5. Verify the increased memory size with the **`QUERY VIRTUAL STORAGE`** command:

```
==> q v stor
00: STORAGE = 1G
```

6. Run the **`SLES11S3 EXEC`** to purge the reader, punch the bootstrap files and IPL from the reader. You should see the Linux RAMdisk getting loaded into memory. Look for the contents of the parameter file you created:

```
==> sles11s3
...
Linux version 3.0.76-0.9-default (geeko@buildhost) (gcc version 4.3.4 [gcc-4_3-b
ranch revision 152973] (SUSE Linux) ) #1 SMP Fri May 31 09:17:47 UTC 2013 (82ad5
ef)
setup.1a06a7: Linux is running as a z/VM guest operating system in 64-bit mode
Zone PFN ranges:
  DMA      0x00000000 -> 0x00080000
  Normal   empty
Movable zone start PFN for each node
early_node_map[1] active PFN ranges
    0: 0x00000000 -> 0x00040000
PERCPU: Embedded 10 pages/cpu @0000000002a4d000 s11776 r8192 d20992 u40960
Built 1 zonelists in Zone order, mobility grouping on.  Total pages: 258560
Kernel command line: ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dum
b                    HostIP=9.12.7.8       Hostname=virtcook8.itso.ibm.com
                     Gateway=9.12.4.1       Netmask=255.255.240.0 Layer2=1
                     ReadChannel=0.0.0600  WriteChannel=0.0.0601  DataChannel=0.
0.0602               Nameserver=9.12.6.7        portname=whatever portno=0
                     Install=nfs://9.12.5.251/srv/nfs/s11s3
                     UseSSH=1 SSHPassword=12345678
                     UseVNC=1  VNCPassword=12345678
                     InstNetDev=osa OsaInterface=qdio OsaMedium=eth Manual=0
...
```

7. You should be prompted for the MAC address. Press **Enter** twice:

```
MAC address
Enter Enter
```

8. You should be prompted for the IP address of your name server. Press **Enter** twice:

```
Enter the IP address of your name server. Leave empty or enter "+++" if you
don't need one
[9.12.6.7]> Enter Enter
```

9. You should see messages showing the installation system being loaded. For example:

```
...
Loading Installation System (1/6) (24640 kB) -          0%       1%       2%
 3%       4%       5%       6%       7%       8%       9%      10%      11%      12%
13%      14%      15%      16%      17%      18%      19%      20%      21%      22%
...
```

If you do not see these, verify all is well with the NFS server: Try mounting the same NFS file system from a different server, be sure that a firewall is not running, verify the syntax in the `PARM-S11` file, and so on.

10. A VNC server process will be started. You should see the following messages:

```
starting VNC server...
A log file will be written to: /var/log/YaST2/vncserver.log ...

***
***           You can connect to <host>, display :1 now with vncviewer
***           Or use a Java capable browser on http://<host>:5801/
***
*** Starting YaST2 ***
```

11. Start a VNC viewer session to the installation process. In this example, a RealVNC client connects to ***9.12.7.8*:1** as shown on the left side of Figure 15-1. Enter the password specified in the parameter file (**12345678** in this example) as shown on the right.

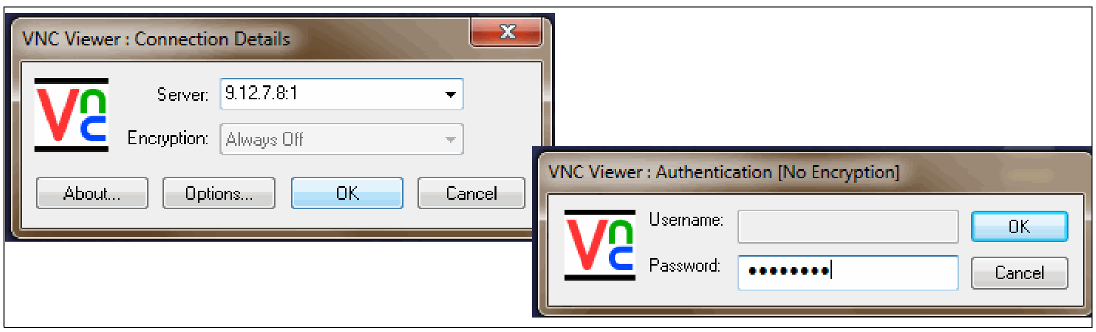


Figure 15-1 Using the VNC viewer

12. You should see a window entitled *Welcome*. Select your *Language* and *Keyboard*. After reading the License Agreement, click the check-box **I Agree to the License Terms**, then click **Next**.
13. At the *Disk Activation* window, click **Configure DASD Disks**.
14. At the *DASD Disk Management* window:
    a. Click **Select or Deselect** for the four read/write disks (**100, 101, 300,** and **301**).
    b. Click **Activate** on the *Perform Action* pop-up menu. If the DASDs have not been formatted previously, you will be asked if you want to format them now. Format the DASDs now. This will take a few minutes.
    c. If you want to make sure to get the disks in a certain device order, activate them one by one. The first disk will be **/dev/dasda**, the second disk will be **/dev/dasdb**, and so on.
    d. The DASDs will be activated quickly, as shown on Figure 15-2 on page 265.

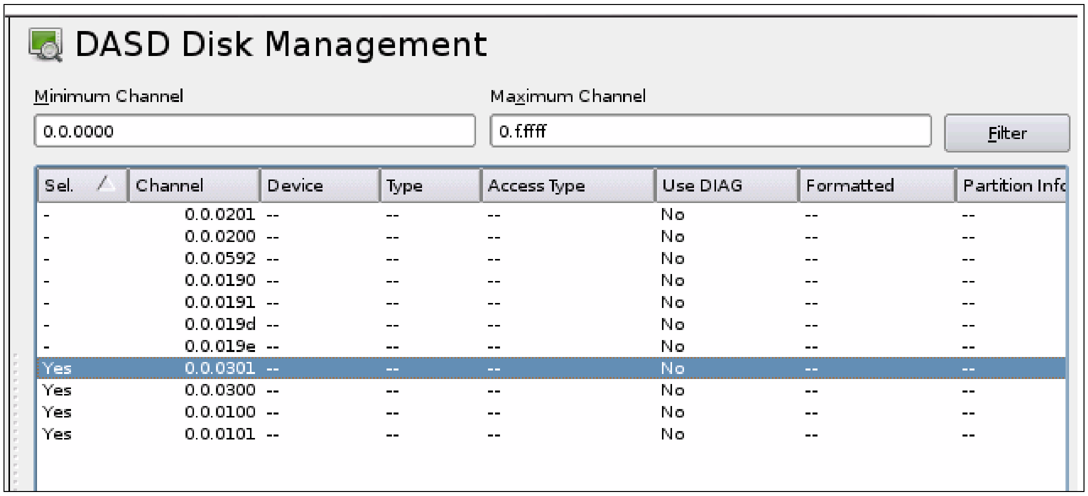


Figure 15-2 Activating DASD on the Linux administration system

   e. Click **Next** when the formatting is complete.
   f. In the *Disk Activation* window, click **Next**.

15. If you see a pop-up window stating “The partitioning on disk ***X*** is not readable by the partitioning tool parted...”, click **OK**.
16. From this point, the installation process behaves identically to any other architecture.
17. In the *Installation Mode* window, accept the default of **New installation** and click **Next**.
18. In the *Clock and Time Zone* window, select your time settings and click **Next**.
19. In the *Installation Settings* window, click the tab **Expert**.
20. In the *Installation Settings* window, click **Partitioning**.
21. In the *Preparing Hard Disk* window, accept the default of **Customer Partitioning (for experts)** and click **Next**.
22. In the *Expert Partitioner* window and the *System View* column on the left, click a plus sign next to **Hard Disks** to expand a list of all available disks, as shown in Figure 15-3 on page 266.

> **RHEL or SLES?:** If you are working only with SLES 11 SP3, you should apply the steps in this chapter to the `LNXADMIN` virtual machine on each member. When installing on member 1, add a logical volume mounted over `/srv/`.

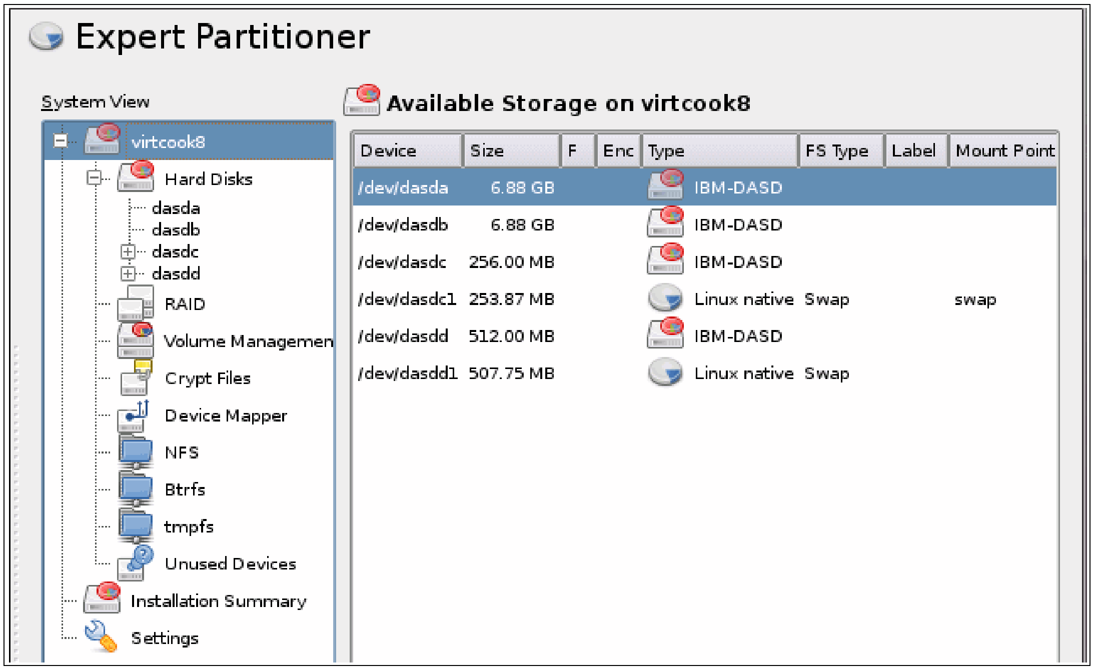


Figure 15-3 Expert partitioner: Hard Disks

23. Double-click the **IBM-DASD** that corresponds to the `100` minidisk in the *Available Storage* section. In this example, it is `/dev/dasda`.
24. In the *Hard Disk: /dev/dasda* section, click the **Add** button to add a partition.
25. In the *Add Partition* window on */dev/dasda*, set partition size to **4 G** and click **Next**.
26. On the next screen, accept the defaults (Format partition, Ext3 file system, Mount partition, and a *Mount Point* of `/`), as shown in Figure 15-4. Click **Finish**.

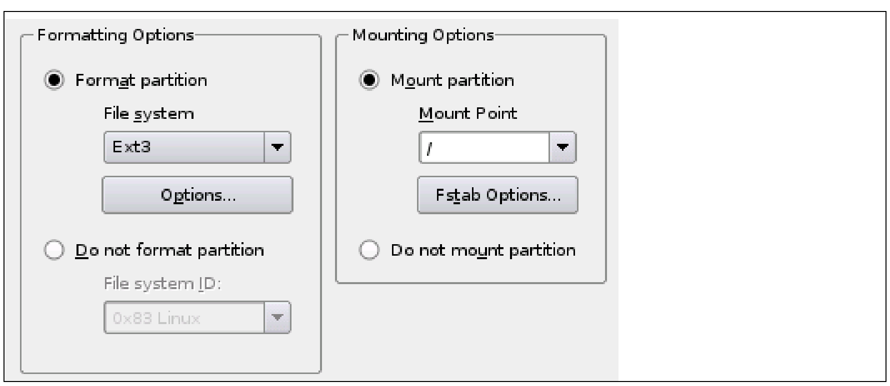


Figure 15-4 Add partition on /dev/dasda

27. Define the rest of the first **DASD** and the second **DASD** as physical volumes for a logical volume:
    - The **dasda** disk should be selected. Click **Add**.
    - In the *Add Partition* window on */dev/dasda*, click **Maximum Size**, then click **Next**.
    - In the next pop-up window, select **Do not format partition** on the left, and **Do not mount partition** on the right. Click **Finish**.
    - Back in the *Expert Partitioner* window, select **dasdb** and click **Add**.

– In the *Add Partition* window on `/dev/dasda`, click **Maximum Size**, then **Next**.

– In the next pop-up window, select **Do not format partition** on the left, and **Do not mount partition** on the right. Click **Finish**.

28. The two virtual disks, `/dev/dasdc1` and `/dev/dasdd1`, should be recognized as swap devices.

29. To define the order of the swap devices, perform the following steps:

    – Select the smaller swap partition, which should be **dasdc**, and click **Edit.**

    – Click **Fstab Options** and set the **Swap Priority** to **2**. Click **Ok** and **Finish**. This will give the smaller VDisk swap space a higher priority.

    – Select the larger swap partition and set the swap priority to **1**. This will give the larger VDisk swap space a lower priority.

30. Set up a large volume group as follows:

    – On the *System View* on the left side, select **Volume Management**, then click **Add**, then choose **Volume Group** from the drop-down menu.

    – Set the **Volume Group Name** to `lnxadmin-vg`.

    – In the buttons in the center, click **Add All** to add both available physical volumes.

    – Click **Finish**.

31. On the left, **Volume Management** should still be selected. Click **Add**, then choose **Logical Volume** from the drop-down menu.

32. In the *Add Logical volume srv* screen, set the name of the *Logical Volume* to **srv**. Click **Next**.

33. On the next window, accept the default of the **Maximum Size** and click **Next**.

34. On the next window, set the *File System* to **XFS** and the *Mount Point* to **/srv.** Click **Finish**.

35. On the *System View* on the left side, select the Linux system at the top.

36. The *Expert Partitioner* panel should now look as it does in Figure 15-5. When it is correct, click **Accept**.

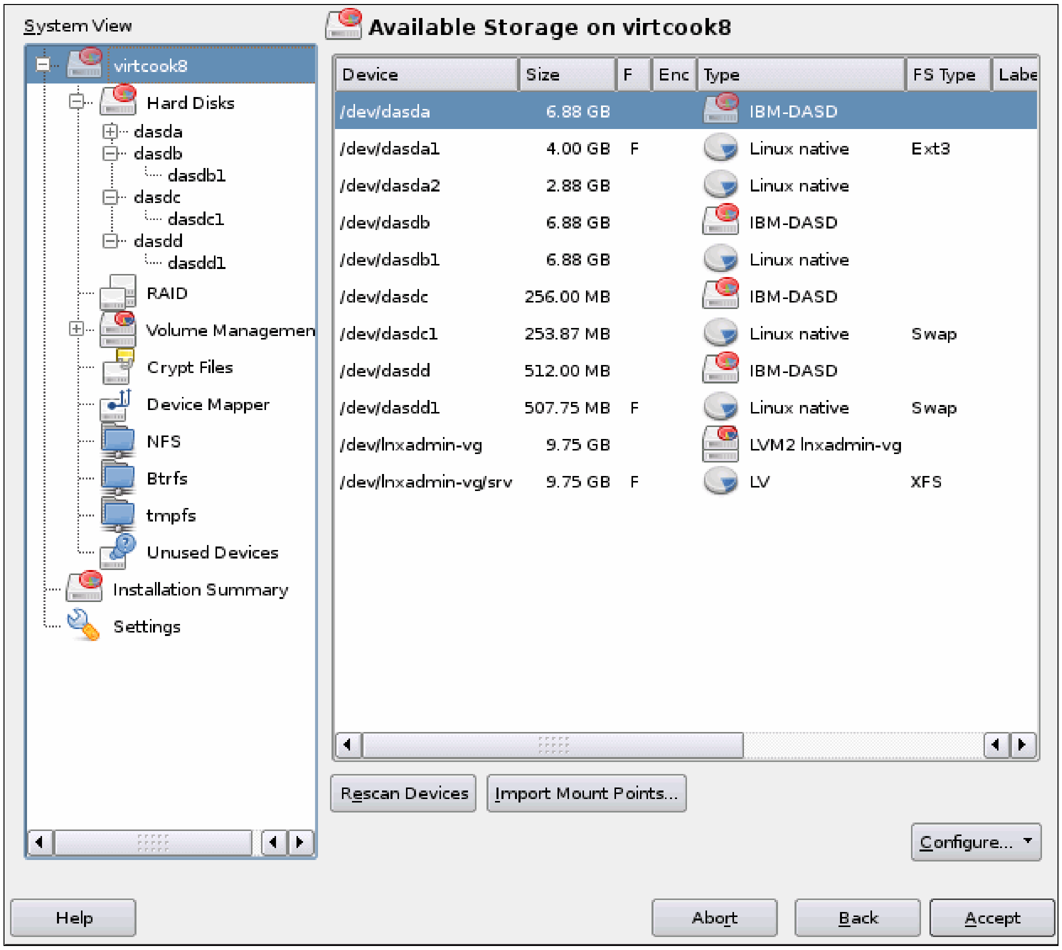


Figure 15-5 Disk allocations in the Expert Partitioner

37. Back on the *Installation Settings* window, click **Software**. The Software Selection and System Tasks window should appear (you might get a warning window about low disk space; this is acceptable).

38. Clear all items except **Base System, 32-Bit Runtime Environment, Help and Support Documentation**, and **Minimal System**, as shown in Figure 15-6. When finished, click **OK**.

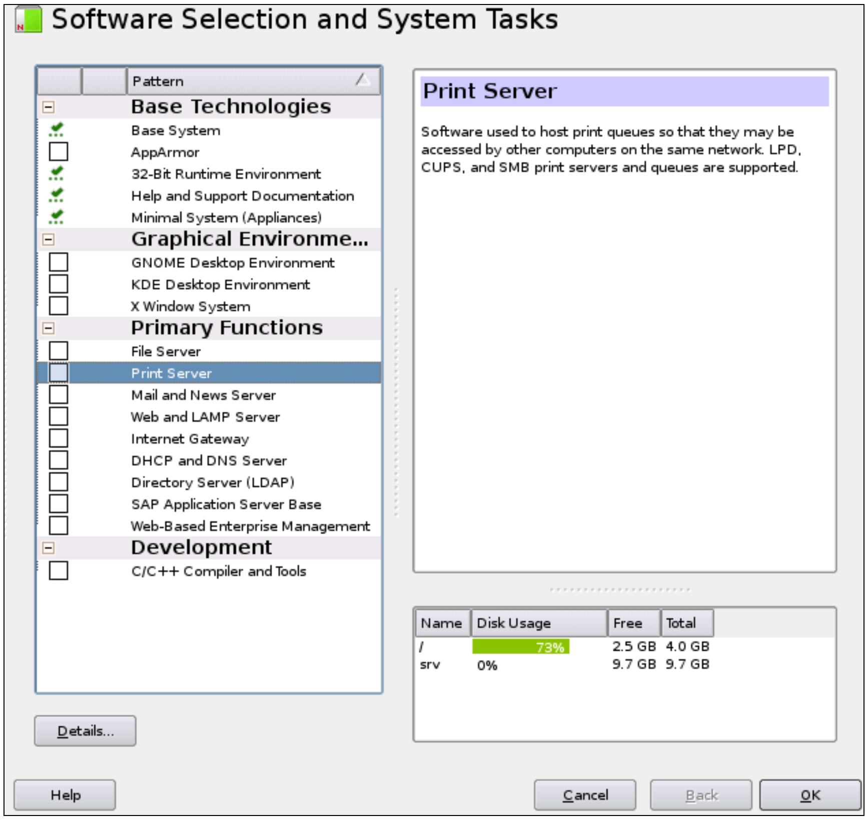


Figure 15-6 Choosing software groups

39. In the *Expert* tab, click **Default Runlevel** near the bottom.

40. Choose **3: Full multiuser with network**, as shown in Figure 15-7. Click **OK**. If you receive a VNC warning, click **Yes**.

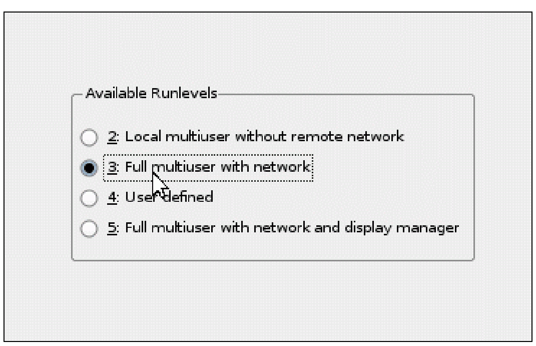


Figure 15-7 Setting default runlevel

41. Click the *Overview* tab again. You should see a summary of the proposed installation, as shown in Figure 15-8.

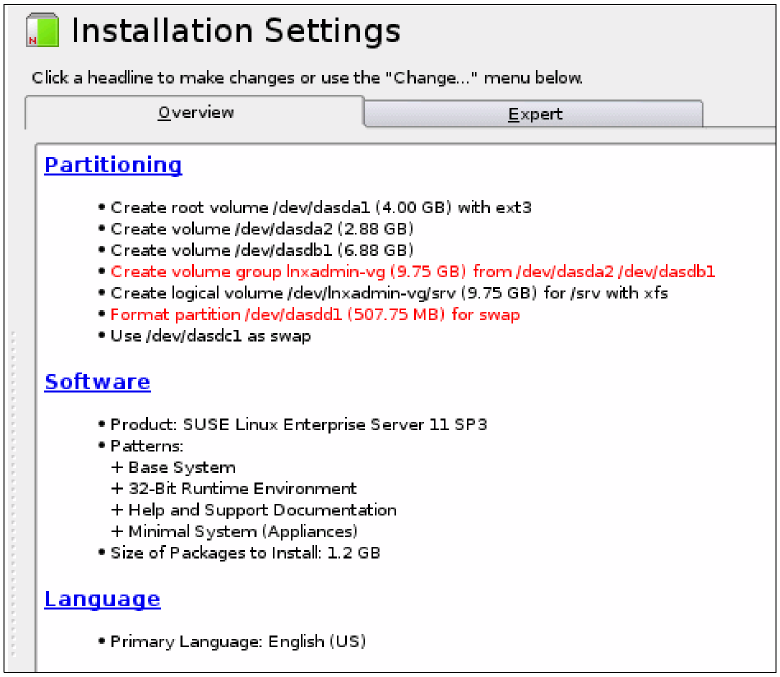


Figure 15-8 Installation Settings summary

42. In the *Installation Settings* window, click **Install**.

43. On the *Confirm Installation* window, click **Install**. This will begin the process of laying down RPMs onto disk. Copying the RPMs should take about 5 - 15 minutes. When copying of the RPMs is done, a few more windows will pass by and then your VNC viewer session will close.

15.3.1 Complete the Linux administration system installation

In order to complete the Linux administration system installation, perform the following steps:

1. Go back to your 3270 session. You will see messages indicating the Linux image is being restarted. You might need to clear the screen a number of times. At the end of the re-IPL, the VNC server is started again:

```
...
***
***  Please return to your X-Server screen to finish installation
***

0

starting VNC server...
A log file will be written to: /var/log/YaST2/vncserver.log ...

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

(When YaST2 is finished, close your VNC viewer and return to this window.)
```

```
Active interfaces:

eth0      Link encap:Ethernet  HWaddr 02:00:0C:00:00:21
          inet addr:9.12.7.8  Bcast:9.12.15.255  Mask:255.255.240.0
--
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0

...
```

2. Start or reconnect the VNC viewer session again by using the same credentials.
3. On the *Password for the System Administrator "root"* window, set the root password twice and click **Next**. You might get a warning about the strength of the password.
4. On the *Hostname and Domain Name* window, the *Hostname* and *Domain Name* fields should be completed by values in the parameter file (`LNXADMIN PARM-S11`). **Clear** the *Change Hostname via DHCP* box, and click **Next**.
5. On the *Network Configuration* window, in the *Firewall* section, click **disable**, which disables the firewall. Click **Next**.
6. On the *Test Internet Connectio*n window, if you do not have an Internet connection, select **No, Skip This Test**. However, SUSE recommends you perform this test if possible. Click **Next**.
7. On the *Network Services Configuration* window, accept the defaults and click **Next**. A certificate will be created.
8. On the *User Authentication Method* window, select **Local (/etc/passwd)** and click **Next**.
9. On the *Add a new local user* window, add a non-root user for the primary system administrator for this system and click **Next**.
10. At the *Release Notes* window, it is recommended that you read the information. Click **Next**.
11. At the *Hardware Configuration* window, click **Next**.
12. At the *Installation Complete* window, **clear** the check box *Clone this system for AutoYaST2*, then click **Finish**. The VNC viewer session will end.
13. Go back to the 3270 session and you might have to clear the screen a few times. When you see the login prompt, **`DISCONNECT`** using the **`#CP`** prefix:

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

You have now installed the Linux administration system. You should be able to access the new system using SSH.

15.3.2 Verify the installation

To verify the installation, perform the following steps:

1. Start an SSH session to the Linux administration system as `root`.
2. Show the file system sizes with the **`df -h`** command:

```
# df -h
Filesystem                    Size  Used Avail Use% Mounted on
/dev/dasda1                   4.0G  1.4G  2.4G  38% /
udev                          498M  120K  498M   1% /dev
tmpfs                         498M     0  498M   0% /dev/shm
/dev/mapper/lnxadmin--vg-srv  9.8G   33M  9.8G   1% /srv
```

3. Verify that there are two swap spaces with the **swapon -s** commands:

```
# swapon -s
Filename                                Type            Size    Used    Priority
/dev/dasdc1                             partition       259956  0       2
/dev/dasdd1                             partition       519924  0       1
```

SUSE Linux Enterprise Server 11 SP3 is now installed on the Linux administration system. The next step is to configure it.

15.4 Configure the SUSE Linux Enterprise Server 11 SP3 Linux administration system

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

1. "Copy installation files to the Linux administration system" on page 272
2. "Configure an FTP server" on page 274
3. "Configure an NFS server" on page 274
4. "Reset the installation source location" on page 275
5. "Turn off unneeded services" on page 276
6. "Apply service if necessary: Online update" on page 277
7. "Install the cmsfs package" on page 277
8. "Enable the vmcp modules" on page 278
9. "Set system to halt on SIGNAL SHUTDOWN" on page 278
10. "Reboot the system" on page 279
11. "Verify the changes" on page 279

15.4.1 Copy installation files to the Linux administration system

In this section, you will copy the SUSE Linux Enterprise Server 11 SP3 installation files to the Linux administration system on member 1.

To do so, perform the following steps:

1. Start an SSH session to LNXADMIN **on member 1**. That is where the large logical volume is mounted over /srv/.
2. Verify that there is enough disk space left:

```
# df -h /srv
Filesystem                    Size  Used Avail Use% Mounted on
/dev/mapper/lnxadmin--vg-srv  9.8G   33M  9.8G   1% /srv
```

   In this example there is 9.8 GB of disk space free.

3. Create the /srv/ftp/iso/ directory and change into it:

```
# mkdir -p /srv/ftp/iso/
# cd /srv/ftp/iso
```

4. Use the **scp** command to copy the SLES 11 SP3 ISO image(s) from the NFS server to this directory. This step might take some time:

```
# scp 9.12.5.251:/srv/nfs/SLE*.iso .
The authenticity of host '9.12.5.251 (9.12.5.251)' can't be established.
RSA key fingerprint is 70:cf:ed:a8:98:4a:46:05:95:d9:72:b1:94:a9:16:25 [MD5].
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '9.12.5.251' (RSA) to the list of known hosts.
root@9.12.5.251's password:
```

```
SLE-11-SP3-SDK-DVD-s390x-GMC-DVD1.iso            100% 2130MB  10.9MB/s   03:16
SLES-11-SP3-DVD-s390x-GMC-DVD1.iso               100% 3149MB  10.8MB/s   04:53
```

5. Check the size of the `/srv/` file system again:

```
# df -h /srv
Filesystem                    Size  Used Avail Use% Mounted on
/dev/mapper/lnxadmin--vg-srv  9.8G  5.2G  4.6G  54% /srv
```

The file system has filled up significantly from copying the two DVD ISO images. The next step is to mount these images

15.4.2 Automatically mount the two disk images

The ISO images of the disks should be set to be mounted loopback automatically at boot time. To accomplish this task, perform the following steps

1. Make two directories for mount points, `/srv/ftp/SLES/` and `/srv/ftp/SDK/`:

```
# cd /srv/ftp
# mkdir SLES SDK
```

2. Make a backup copy of the original `/etc/fstab` file:

```
# cd /etc
# cp fstab fstab.orig
```

3. Add two lines to the `/etc/fstab` file to loopback-mount both ISO images:

```
# vi /etc/fstab
/srv/ftp/iso/SLES-11-SP3-DVD-s390x-GMC-DVD1.iso    /srv/ftp/SLES iso9660 loop,ro 0 0
/srv/ftp/iso/SLE-11-SP3-SDK-DVD-s390x-GMC-DVD1.iso /srv/ftp/SDK iso9660 loop,ro 0 0
```

4. Run the **mount -a** command to read the /etc/fstab file and perform all mounts:

```
# mount -a
```

5. Verify that the two ISO images are now mounted with the following command:

```
# mount | grep iso9660
/srv/ftp/iso/SLES-11-SP3-DVD-s390x-GMC-DVD1.iso on /srv/ftp/SLES type iso9660 (ro)
/srv/ftp/iso/SLE-11-SP3-SDK-DVD-s390x-GMC-DVD1.iso on /srv/ftp/SDK type iso9660 (ro)
```

15.4.3 Copy the files associated with this book

Copy the files associated with this book from the PC NFS server with the following steps:

1. Use the **scp** command to copy the tar files that are associated with this book. In this example, the IP address of the PC NFS server is ***9.12.5.251***:

```
# cd /srv/ftp
# scp 9.12.5.251:/srv/nfs/SG248147.tgz .
root@9.12.5.251's password:
SG248147.tgz                                     100%   40KB  40.1KB/s   00:00
```

2. Untar the archive:

```
# tar xvf SG248147.tgz
SG248147/
SG248147/disclaimer.txt
SG248147/README.txt
SG248147/vm/
SG248147/vm/lnxmaint/
SG248147/vm/lnxmaint/sample.conf-rh6
SG248147/vm/lnxmaint/sample.parm-rh6
SG248147/vm/lnxmaint/profile.exec
```

```
SG248147/vm/lnxmaint/swapgen.exec
SG248147/vm/lnxmaint/sample.parm-s11
SG248147/vm/lnxmaint/rhel64.exec
SG248147/vm/lnxmaint/sles11s3.exec
SG248147/vm/maint/
SG248147/vm/maint/ssicmd.exec
SG248147/vm/maint/callsm1.exec
SG248147/vm/maint/cpformat.exec
SG248147/rhel64/
SG248147/rhel64/clone-1.0-11.s390x.rpm
SG248147/sles11sp3/
SG248147/sles11sp3/linux5.xml
SG248147/sles11sp3/jeos.tgz
SG248147/sles11sp3/boot.clone
SG248147/sles11sp3/clone.sh
```

The files associated with this book should now be available in the `/srv/ftp/SG248147/` directory.

15.4.4 Configure an FTP server

Set up the Linux administration server as an FTP server by performing the following steps:

1. Invoke the **yast** command.
2. On the left side, use the arrow keys to move down to **Network Services**.
3. On the right side, move down to **FTP Server** and press **Enter**.
4. At the *No server package installed screen*, accept the default of **vsftpd** and click **OK**.
5. In the *FTP Start-Up* section on the right side, move the cursor to **When booting** and use the space bar to select it.
6. In the *Switch On and Off* section, select **Start FTP Now**.
7. Move the cursor to **Finish** and press **Enter** to select.
8. This brings you back to the *YaST2 Control Center*. Select **Quit**.

The **vsftpd** FTP server should now be installed and configured.

15.4.5 Configure an NFS server

Set up the Linux administration server as an NFS server by performing the following steps:

1. Invoke the **yast** command.
2. On the left side, use the arrow keys to move down to **Network Services**.
3. On the right side, move down to **NFS Server** and press **Enter**.
4. When asked to install the `nfs-kernel-server`, click **Install**.
5. In the *NFS Server Configuration* menu, `NFS Server` section, use the space bar to select **Start**.
6. Accept the remaining defaults and click **Next**.
7. In the *Directories to Export* window, move the cursor to the *Add Directory* button and press **Enter**.
8. Add the `/srv/ftp` directory to the list and select **OK**.
9. For the *Host Wild Card* and *Options*, accept all defaults and select **OK**.

10. Back on the main panel, move the cursor with the Tab key and select **Finish**.
11. Exit yast by selecting **Quit**.

You have now copied the files that you need for SLES 11 SP3 and exported them with NFS.

15.4.6 Reset the installation source location

When SUSE Linux Enterprise Server 11 SP3 is installed, the location of the *installation source* is remembered. In this case, it is the PC NFS server. Now that the DVD ISO image has been copied from the PC NFS server to the Linux administration system, you can reset the installation source location to point to this system (so LNXADMIN points to itself).

To accomplish this task, perform the following steps:

1. Start an SSH session as root on the Linux administration system on member 1.
2. Invoke the `yast` command
3. Accept the default of **Software** in the left column and using the arrow keys, select **Software Repositories** in the right column and press **Enter**:

```
+----------------------------------------------------------------------+
| YaST2 Control Center                                                 |
+----------------------------------------------------------------------+
|Software            - -Online Update                                  |
|Hardware            - -Software Management                            |
|System              - -Add-On Products                                |
|Network Devices     - -Installation into Directory                    |
|Network Services    - -Media Check                                    |
|Security and Users  - -Online Update Configuration                    |
|Virtualization      - -Patch CD Update                                |
|Support             - -Software Repositories                          |
...
```

4. At the top of the *Configured Software Repositories* panel, you should see a single entry for SUSE Linux Enterprise Server 11 SP3. Delete this entry first by using the *Tab* key to move to the **Delete** button.
5. Accept the default of **Yes** when asked to confirm.
6. Use the *Tab* key to move to **Add** at the bottom and press **Enter**.
7. Move the cursor down to **FTP** and press the **space bar** to select. Use the Tab key to move to **Next** at the bottom and press **Enter**.
8. In the *Server Directory* window, set **Server Name** to the host name or IP address. In this example, it is ***virtcook8***.
9. In the *Directory on Server* field, set the value to **`SLES`** and click **Next** (because the home directory of the `FTP` user is `/srv/ftp/`, the target directory is relative to that).
10. Accept the License agreement by using the space bar to select **Yes** and clicking **Next**.
11. Select **Add** again, and add the SDK by setting the value to **`SDK`**.
12. Back at the *Configured Software Repositories* panel, you should see the two repositories you just added. Click **OK**.
13. Leave yast by moving to **Quit**.
14. To check if everything is correct, run the ***zypper refresh*** command on the command line:

```
# zypper ref
Retrieving repository 'SUSE-Linux-Enterprise-Server-11-SP3 11.3.3-1.138' metadata [done]
```

```
Retrieving repository 'SUSE-Linux-Enterprise-Software-Development-Kit-11-SP3
11.3.3-1.69' metadata [done]
All repositories have been refreshed.
```

You have now changed the Linux administration system point to the repository on itself. The next chapter tests this by installing the golden image from this repository. When that succeeds, you should be able to retire the PC NFS server because all necessary files are on System z.

15.4.7 Turn off unneeded services

There are a number of services that are started in a SUSE Linux Enterprise Server 11 SP3 minimal system. Some of these can safely be turned off. To do so, perform the following steps:

1. View which services are on in run level 3 with the following **chkconfig** command:

```
# chkconfig -l | grep 3:on
auditd                    0:off  1:off  2:off  3:on   4:off  5:on   6:off
cron                      0:off  1:off  2:on   3:on   4:off  5:on   6:off
dbus                      0:off  1:off  2:on   3:on   4:off  5:on   6:off
earlysyslog               0:off  1:off  2:on   3:on   4:off  5:on   6:off
fbset                     0:off  1:on   2:on   3:on   4:off  5:on   6:off
haldaemon                 0:off  1:off  2:on   3:on   4:off  5:on   6:off
haveged                   0:off  1:off  2:on   3:on   4:off  5:on   6:off
irq_balancer              0:off  1:on   2:on   3:on   4:off  5:on   6:off
network                   0:off  1:off  2:on   3:on   4:off  5:on   6:off
network-remotefs          0:off  1:off  2:on   3:on   4:off  5:on   6:off
nfs                       0:off  1:off  2:off  3:on   4:off  5:on   6:off
nfsserver                 0:off  1:off  2:off  3:on   4:off  5:on   6:off
nscd                      0:off  1:off  2:off  3:on   4:off  5:on   6:off
postfix                   0:off  1:off  2:off  3:on   4:off  5:on   6:off
purge-kernels             0:off  1:off  2:off  3:on   4:off  5:on   6:off
random                    0:off  1:off  2:on   3:on   4:off  5:on   6:off
rpcbind                   0:off  1:off  2:off  3:on   4:off  5:on   6:off
smartd                    0:off  1:off  2:on   3:on   4:off  5:on   6:off
splash                    0:off  1:on   2:on   3:on   4:off  5:on   6:off  S:on
splash_early              0:off  1:off  2:on   3:on   4:off  5:on   6:off
sshd                      0:off  1:off  2:off  3:on   4:off  5:on   6:off
syslog                    0:off  1:off  2:on   3:on   4:off  5:on   6:off
xinetd                    0:off  1:off  2:off  3:on   4:off  5:on   6:off
```

2. Turn off the following services by using the **chkconfig** command:

```
# chkconfig fbset off
# chkconfig network-remotefs off
# chkconfig postfix off
# chkconfig splash off
# chkconfig splash_early off
# chkconfig smartd off
# chkconfig xinetd off
```

3. Review which services are now configured to start in run level 3 with the following command:

```
# chkconfig -l | grep 3:on
auditd                    0:off  1:off  2:off  3:on   4:off  5:on   6:off
cron                      0:off  1:off  2:on   3:on   4:off  5:on   6:off
dbus                      0:off  1:off  2:on   3:on   4:off  5:on   6:off
earlysyslog               0:off  1:off  2:on   3:on   4:off  5:on   6:off
haldaemon                 0:off  1:off  2:on   3:on   4:off  5:on   6:off
haveged                   0:off  1:off  2:on   3:on   4:off  5:on   6:off
```

```
irq_balancer             0:off  1:on   2:on   3:on   4:off  5:on   6:off
network                  0:off  1:off  2:on   3:on   4:off  5:on   6:off
nfs                      0:off  1:off  2:off  3:on   4:off  5:on   6:off
nfsserver                0:off  1:off  2:off  3:on   4:off  5:on   6:off
nscd                     0:off  1:off  2:off  3:on   4:off  5:on   6:off
purge-kernels            0:off  1:off  2:off  3:on   4:off  5:on   6:off
random                   0:off  1:off  2:on   3:on   4:off  5:on   6:off
rpcbind                  0:off  1:off  2:off  3:on   4:off  5:on   6:off
sshd                     0:off  1:off  2:off  3:on   4:off  5:on   6:off
syslog                   0:off  1:off  2:on   3:on   4:off  5:on   6:off
vsftpd                   0:off  1:off  2:off  3:on   4:off  5:on   6:off
```

15.4.8 Apply service if necessary: Online update

You might want to apply service using Yast Online Update. Internet access was not available during the writing of this book, so step-by-step details are not available.

If you have access to the Internet, or an online update source, invoke **yast** → **Software** → **Online update**.

15.4.9 Install the cmsfs package

The **clone.sh** script requires the **cmsfs** package, written by Rick Troth, in order read CMS files. To do so, perform the following steps:

1. Install cmsfs using the **zypper install** command:

```
# zypper install cmsfs
...
Continue? [y/n/?] (y): y
Retrieving package cmsfs-1.1.8-6.4.1.s390x (1/1), 34.0 KiB (148.0 KiB unpacked)
Retrieving: cmsfs-1.1.8-6.4.1.s390x.rpm [done]
Installing: cmsfs-1.1.8-6.4.1 [done]
```

2. To test that the cmsfs package is properly installed, see if you can read the LNXADMIN PARM-S11 parameter file. Bring the 191 disk online using the **chccwdev -e** command:

```
# chccwdev -e 191
Setting device 0.0.0191 online
Done
```

3. View the DASD with the **lsdasd** command:

```
# lsdasd
Bus-ID     Status      Name      Device  Type  BlkSz  Size      Blocks
==============================================================================
0.0.0100   active      dasda     94:0    ECKD  4096   7042MB    1802880
0.0.0101   active      dasdb     94:4    ECKD  4096   7042MB    1802880
0.0.0300   active      dasdc     94:8    FBA   512    256MB     524288
0.0.0301   active      dasdd     94:12   FBA   512    512MB     1048576
0.0.0191   active      dasde     94:16   ECKD  4096   351MB     90000
```

This shows that the 191 disk is /dev/dasde.

4. Test the **cmsfslst** command to list the files on the 191 disk:

```
# cmsfslst -d /dev/dasde
FILENAME FILETYPE FM FORMAT LRECL       RECS     BLOCKS     DATE     TIME
         DIRECTOR P0 F         64         18          1  6/06/2013 13:41:39
         ALLOCMAP P0 F       4096          3          3  6/06/2013 13:41:39
LNXADMIN PARM-S11 D1 V         57         10          1  6/06/2013 10:38:19
PROFILE  EXEC     B1 V         63         17          1  6/04/2013 16:35:58
```

```
...
```

5. Test the **cmsfscat** command to list the contents of the file:

```
# cmsfscat -d /dev/dasde -a lnxadmin.parm-s11
ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dumb
HostIP=9.12.7.8       Hostname=virtcook8.itso.ibm.com
...
```

You should now have cmsfs installed and tested.

15.4.10 Enable the vmcp modules

The **vmcp** module/command allows z/VM CP commands to be issued from Linux. It is critical to the functioning of the **clone.sh** script.

To configure the **vmcp** module to be loaded at boot time, perform the following steps:

1. Change the directory to /etc/sysconfig/:

```
# cd /etc/sysconfig
```

2. Make a backup of the original file:

```
# cp kernel kernel.orig
```

3. Edit the kernel file and add the module names to the variable MODULES_LOADED_ON_BOOT (around line 30):

```
# vi kernel
...
## Type:               string
## ServiceRestart:     boot.loadmodules
#
# This variable contains the list of modules to be loaded
# once the main filesystem is active
# You will find a few default modules for hardware which
# can not be detected automatically
#
MODULES_LOADED_ON_BOOT="vmcp"
...
```

4. Save the file.

You should be able to issue CP commands by using the **vmcp** command after your system is rebooted.

15.4.11 Set system to halt on SIGNAL SHUTDOWN

The Ctrl-Alt-Del key sequence is simulated by z/VM when it issues a **SIGNAL SHUTDOWN** command. Rather than rebooting, you want your system to halt (shut down). Change this setting by changing **shutdown -r** to **shutdown -h** in the /etc/inittab file:

```
# cd /etc
# cp inittab inittab.orig
# vi inittab    // change shutdown -r to shutdown -h
...
# what to do when CTRL-ALT-DEL is pressed
ca::ctrlaltdel:/sbin/shutdown -h -t 4 now
...
```

This change will be picked up when the system is rebooted.

15.4.12 Reboot the system

You should now reboot the system to test the changes:

```
# reboot

Broadcast message from root (pts/0) (Wed Oct 19 16:03:44 2011):

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

Your system should be back in a few minutes. You are now done customizing the Linux administration system Linux image.

15.4.13 Verify the changes

To verify the changes, perform the following steps:

1. Start an SSH session as **root** to the Linux administration system.
2. Test the **vmcp** command with the CP **QUERY NAMES** command:

```
# vmcp q n
LINUX2   - SSI , VMSERVP  - SSI
TCPIP    - DSC , DTCVSW2  - DSC , DTCVSW1  - DSC , VMSERVR  - DSC
VMSERVU  - DSC , VMSERVS  - DSC , OPERSYMP - DSC , DISKACNT - DSC
EREP     - DSC , FTPSERVE - DSC , PERFSVM  - DSC , OPERATOR -L0003
LNXADMIN - DSC
VSM     - TCPIP
```

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

```
# swapon -s
Filename                                Type            Size    Used    Priority
/dev/dasdc1                             partition       259956  0       2
/dev/dasdd1                             partition       519924  0       1
```

4. Confirm that the FTP service is running.

```
# service vsftpd status
Checking for service vsftpd                                              running
```

Congratulations. You have installed and configured a SUSE Linux Enterprise Server 11 SP3 Linux system onto the Linux administration system. The next step is to install and configure the SUSE Linux Enterprise Server 11 SP3 golden image.

16

Install the SUSE Linux Enterprise Server 11 SP3 golden image

"I never think of the future. It comes soon enough."

— Albert Einstein

Red Hat Enterprise Linux or SUSE Linux Enterprise Server?: If you are working only with RHEL 6.4, you can skip this chapter because the RHEL 6.4 golden image should be installed.

This chapter describes how to install SUSE Linux Enterprise Server 11 SP3 onto the virtual machine `S113GOLD`, which is referred to as the *golden image*. The golden image is the copy of Linux that will be cloned. Normally the system is shut down and the virtual machine logged off because it is not recommended to clone a running Linux system.

In this example, the golden image is given two 3390 minidisks with 5008 cylinders each at addresses `100` and `101`. This allows for about 7 GB of disk space for each Linux system. If you want to increase that size, larger volumes such as 3390-9s can be used to give approximately 14 GB. However, but minidisks at addresses `100` and `101` must still be defined for the **`clone.sh`** script to function.

To install and configure the golden image, perform the following steps:

1. "Create the S113GOLD virtual machine" on page 281
2. "Create the S113GOLD parameter file" on page 282
3. "Install the golden image" on page 283
4. "Configure the golden image" on page 295

16.1 Create the S113GOLD virtual machine

The golden image has a default memory size of 512 MB, and it is given class G privilege. It is given the following minidisks:

`100` Half of the disk space for the golden image.

101 The other half of the disk space.

To define the S113GOLD virtual machine, perform the following steps:

1. **Log on to** MAINT.
2. Edit the USER DIRECT file, add six new lines at the bottom of the file and create the following user directory entry. Set the 3390 disk labels to those appropriate for your system. In this example, ... is used:

```
==> x user direct c
====> bot
====> a 6
*
USER S113GOLD LNX4VM 512M 1G G
 INCLUDE LNXDFLT
 OPTION LNKNOPAS APPLMON
MDISK 100 3390 0001 5008 JM1265 MR LNX4VM LNX4VM LNX4VM
MDISK 101 3390 5009 5008 JM1265 MR LNX4VM LNX4VM LNX4VM
```

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

```
==> directxa user
z/VM USER DIRECTORY CREATION PROGRAM - VERSION 6 RELEASE 2.0
EOJ DIRECTORY UPDATED AND ON LINE
HCPDIR494I User directory occupies 107 disk pages
```

You have now defined the virtual machine that will contain the Linux golden image.

16.2 Create the S113GOLD parameter file

A SUSE Linux Enterprise Server 11 SP3 parameter file will be needed for this new virtual machine. You will need to change the IP address (HostIP variable) and the host name (Hostname variable). In this example, those are ***9.12.7.10*** and ***virtcook10***.

Also, the Linux administration system is used to provide the SUSE Linux Enterprise Server 11 SP3 install directory using NFS by setting the Install variable.

Perform the following steps:

1. Log on to LNXMAINT.
2. Copy the LNXADMIN parameter file to one with a file name of S113GOLD on the LNXMAINT 192 (D) disk:

```
==> copy lnxadmin parm-s11 d s113gold = =
```

3. Edit the new file and set the networking values correctly. These changes set the golden image's IP address and host name, and also points to the new installation server on the Linux administration system on member 1 (***9.12.7.10*** in this example):

```
==> x s113gold parm-s11
ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dumb
HostIP=9.12.7.10        Hostname=virtcook10.itso.ibm.com
Gateway=9.12.4.1        Netmask=255.255.240.0 Layer2=1
ReadChannel=0.0.0600  WriteChannel=0.0.0601  DataChannel=0.0.0602
Nameserver=9.12.6.7        portname=whatever portno=0
Install=nfs://9.12.5.251/srv/nfs/s11s3
UseSSH=1 SSHPassword=12345678
UseVNC=1  VNCPassword=12345678
InstNetDev=osa OsaInterface=qdio OsaMedium=eth Manual=0
```

4. Log off from LNXMAINT.

When S113GOLD is logged on to, the new parameter file will be accessible on the A (191) disk.

16.3 Install the golden image

You should now be ready to begin the installation onto the golden image. Linux will be installed onto the 100-101 minidisks. It will use 300-301 virtual disks for swapping. Most Linux virtual machines described in this book will have two read/write minidisks and two virtual disks. Disk 300 is 256 MB and will act as a primary swap space. Only after it is full, disk 301, which is 512 MB, will be used.

To install the golden image, perform the following steps:

1. Log on to S113GOLD. When you log on, you should see messages indicating that a virtual NIC has been created starting at address 0600 and that virtual disks 300 and 301 have been created:

```
00: z/VM Version 6 Release 3.0, Service Level 0000 (64-bit),
00: built on IBM Virtualization Technology
00: There is no logmsg data
00: FILES:   NO RDR,   NO PRT,   NO PUN
00: LOGON AT 12:21:42 EDT FRIDAY 06/07/13
00: Command complete
00: NIC 0600 is created; devices 0600-0602 defined
00: NIC 0600 is connected to VSWITCH SYSTEM VSW1
z/VM V6.3.0    2012-08-21 20:25

DMSACP723I A (191) R/O
DMSACP723I C (592) R/O
DIAG swap disk defined at virtual address 300 (64988 4K pages of swap space)
DIAG swap disk defined at virtual address 301 (129980 4K pages of swap space)
Do you want to IPL Linux from minidisk 100? y/n
```

2. You are prompted to IPL Linux, but since you have not installed Linux yet, answer **n**:

```
Do you want to IPL Linux from minidisk 100? y/n
==> n
```

3. Use the **DEFINE STORAGE** command to move the memory size up to 1 GB (512 MB is enough to complete the installation process, but 1 GB will be faster), then re-IPL CMS:

```
==> def stor 1g
00: STORAGE = 1G
==> ipl cms
z/VM V6.2.0    2011-07-19 16:53

DMSACP723I A (191) R/O
DMSACP113S C(592) not attached or invalid device address
DIAG swap disk defined at virtual address 300 (64989 4K pages of swap space)
DIAG swap disk defined at virtual address 301 (129981 4K pages of swap space)
Do you want to IPL Linux from minidisk 100? y/n
n
```

This shows that you have the resources necessary to install SUSE Linux Enterprise Server 11 SP3.

16.3.1 Begin the SUSE Linux Enterprise Server 11 SP3 installation

Follow these steps to begin the installation of S113GOLD.

1. Run the **sles11s3 EXEC**. You should see many panels of questions and answers scrolling by. If you had used the default parameter file shipped with SUSE Linux Enterprise Server 11 SP3, you would have had to answer all the networking questions manually. With the proper parameters set in the file S113GOLD PARM-S11, the install process should proceed to where you access the installation program using a VNC client:

```
==> sles11s3
00: 0000003 FILES PURGED
00: RDR FILE 0038 SENT FROM S113GOLD PUN WAS 0038 RECS 113K CPY  001 A NOHOLD NO
KEEP
00: RDR FILE 0042 SENT FROM S113GOLD PUN WAS 0042 RECS 0009 CPY  001 A NOHOLD NO
KEEP
00: RDR FILE 0046 SENT FROM S113GOLD PUN WAS 0046 RECS 204K CPY  001 A NOHOLD NO
KEEP
00: 0000003 FILES CHANGED
00: 0000003 FILES CHANGED
Initializing cgroup subsys cpuset
Initializing cgroup subsys cpu
Linux version 3.0.76-0.9-default (geeko@buildhost) (gcc version 4.3.4 [gcc-4_3-b
ranch revision 152973] (SUSE Linux) ) #1 SMP Fri May 31 09:17:47 UTC 2013 (82ad5
ef)
setup.1a06a7: Linux is running as a z/VM guest operating system in 64-bit mode
Zone PFN ranges:
  DMA      0x00000000 -> 0x00080000
  Normal   empty
Movable zone start PFN for each node
early_node_map[1] active PFN ranges
    0: 0x00000000 -> 0x00040000
PERCPU: Embedded 10 pages/cpu @0000000002a4d000 s11776 r8192 d20992 u40960
Built 1 zonelists in Zone order, mobility grouping on.  Total pages: 258560
Kernel command line: ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dum
b                    HostIP=9.12.7.10        Hostname=virtcook10.itso.ibm.com
                     Gateway=9.12.4.1        Netmask=255.255.240.0 Layer2=1
                     ReadChannel=0.0.0600  WriteChannel=0.0.0601  DataChannel=0.
0.0602               Nameserver=9.12.6.7        portname=whatever portno=0
                     Install=ftp://9.12.7.8/SLES
                     UseSSH=1 SSHPassword=12345678
                     UseVNC=1  VNCPassword=12345678
                     InstNetDev=osa OsaInterface=qdio OsaMedium=eth Manual=0
...
```

2. Press the **Enter** key twice when prompted for MAC address and the IP address of the name server:

```
...
MAC address
>
...
Enter the IP address of your name server. Leave empty or enter "+++" if you
don't need one
[9.12.6.7]>
...
```

3. You should next see the installation system loading:

```
Loading Installation System (1/6) (24640 kB) -          0%      1%      2%
 3%      4%      5%      6%      7%      8%      9%     10%     11%     12%
...
```

If you do not, verify that all is correct with the FTP server on LNXADMIN.

4. The installation system or *starter system* should continue to boot. You should see the message:

```
starting VNC server...
A log file will be written to: /var/log/YaST2/vncserver.log ...

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

(When YaST2 is finished, close your VNC viewer and return to this window.)

Active interfaces:

eth0      Link encap:Ethernet  HWaddr 02:00:0C:00:00:20
          inet addr:9.60.18.145  Bcast:9.60.18.255  Mask:255.255.255.128
--
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0

*** Starting YaST2 ***
```

5. Use a VNC viewer through a Java enabled browser, or a stand-alone VNC viewer. Connect to the VNC server (***9.12.7.10:1*** in this example). Enter the password that is specified in the parameter file (***12345678*** in this example).
6. You could disconnect from the 3270 session; however, messages to the console will be lost. It is recommended that you stay connected, but you might have to clear the window periodically (or the installation process might be delayed waiting for the window to clear itself).

Now the graphical installation process starts.

16.3.2 Begin YaST installation

Perform the following steps to install SUSE Linux Enterprise Server 11 SP3:

1. On the *Welcome* panel, choose your language and keyboard (**English US** in this example). Read the License Agreement, choose **I Agree to the License Terms** and click **Next**.
2. The *Disk Activation* window should appear. Choose **Configure DASD Disks**.
3. The *DASD Disk Management* window should appear. You will see all the DASD available to S113GOLD:
   a. Highlight each of the minidisks and virtual disks, 100, 101, 300 and 301 and click **Select or Deselect**.
   b. You should see a **Yes** appear next to them in the *Sel.* column on the left. Activate them by clicking **Perform Action** -> **Activate**, as shown on the left side of Figure 16-1 on page 286.
   c. Disks 100 and 101 must be formatted so that Linux can use them. Deselect disks 300 and 301 using the **Select or Deselect** button so that 100 and 101 remain selected. Now click **Perform Action** -> **Format** as shown on the right side of the figure.

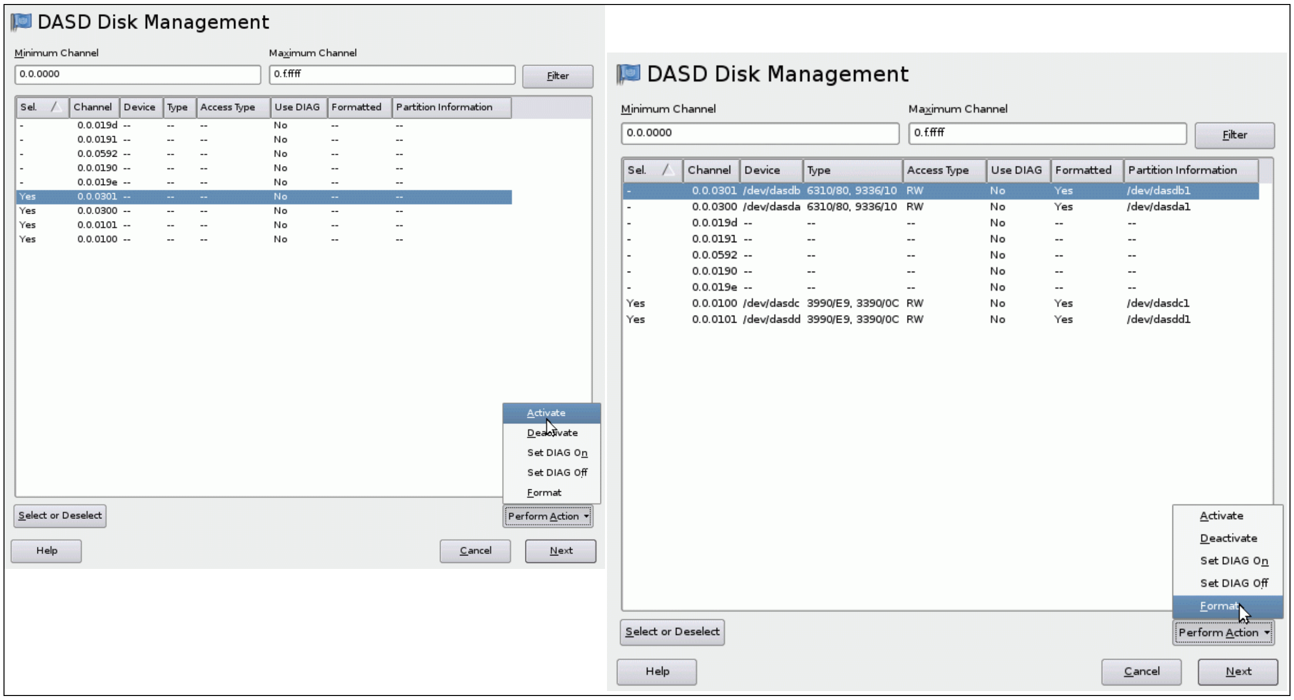


Figure 16-1 DASD available to S113GOLD

4. You should see a window asking for *2 Parallel Formatted Disks*. Click **OK**.
5. Click **Yes** to the question *Really format the following disks?*
6. A progress indicator window should appear displaying the progress of the parallel formats. This step can take 2 - 10 minutes depending on a number of factors.
7. When the formatting is complete, click **Next** in the *DASD Disk Management* window.
8. In the *Disk Activation* window, click **Next** again.
9. In the *Installation Mode* window, accept the default of **New installation** and click **Next**.
10. The *Clock and Time Zone* window will appear. Choose your region and time zone and click **Next**.
11. This will bring you to the *Installation Settings* window. Click **Partitioning** in the *Overview* tab. The *Preparing Hard Disk* window will appear.
12. Accept the default of **Custom Partitioning (for experts)** and click **Next**. The *Expert Partitioner* window appears, as shown in Figure 16-2 on page 287.

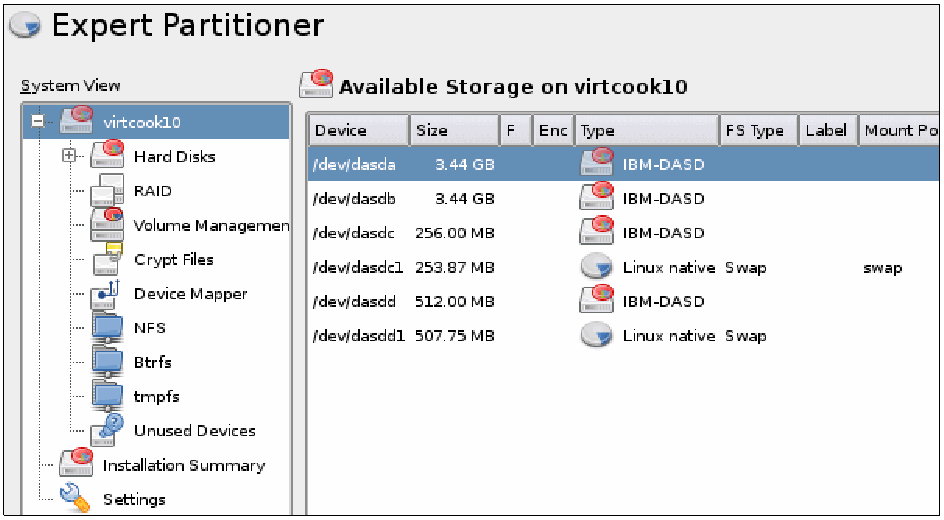


Figure 16-2 Disk partitioner: Before customization

13. Two partitions will be made on `/dev/dasda` and two partitions will be made on `/dev/dasdb`:
    a. Click the plus sign (**+**) to the left of **Hard Disk**. It should show `dasda-dasdd` in the tree.
    b. Click **dasda** on the left and then on **Add**. The window *Add Partition on /dev/dasd*a will appear.
14. Accept the default of the **Custom size** radio button and set the size of **800 MB,** as shown in Figure 16-3. Click **Next**.

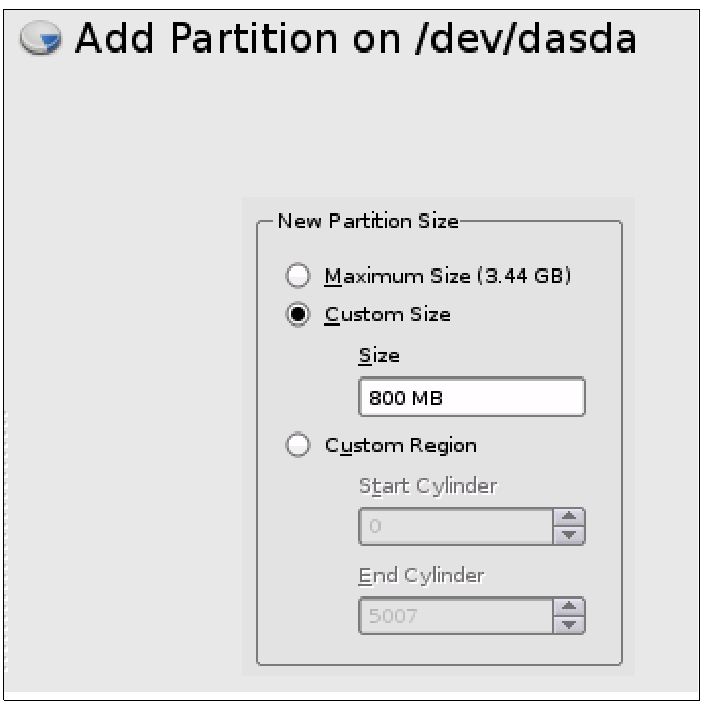


Figure 16-3 Add Partition on /dev/dasda panel

15. You should see a new window: *Add Partition on /dev/dasda:*
    – Accept the default of **Format Partition**.
    – Accept the default of a *File system* of type **ext3** in the *Formatting Options* section.

– Accept the default of a *Mount Point* of **/** (root file system) in the *Mounting Options* section.

– Click **Finish**. This will create the partition `/dev/dasda1`.

16. To create a second partition click **Add** while `dasda` is selected in the left tree. The window *Add partition on /dev/dasda* will appear.
17. Click the **Maximum Size** radio button then click **Next**.
18. In new *Add partition on /dev/dasda* window as shown in Figure 16-4, perform the following steps:
    a. Click the **Do not format** radio button in the *Formatting Options* section.
    b. Click **Do not mount partition** radio button in the *Mounting options* section.
    c. Click **Finish**. This will create an empty partition: `/dev/dasda2`.

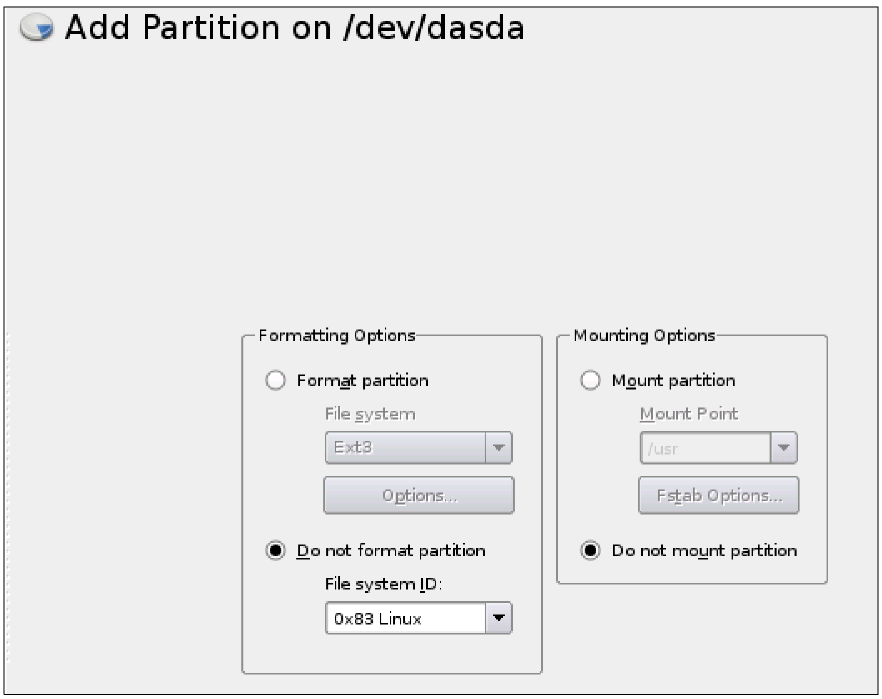


Figure 16-4 Add Partition on /dev/dasda

19. Create the partition `/dev/dasdb1` in the same fashion as you did to create `/dev/dasda1` (Select `dasdb` => **Add** => Custom Size 800 MB => **Format with ext3**). Use **`/tmp/`** as the mount point.
20. Create an empty partition `/dev/dasdb2` in the same fashion as you did to create `/dev/dasda2` (Select `/dev/dasdb` => **Add** => Maximum Size => **Do not format/Do not mount partition**). Accept the default size of all the space.
21. Now `/dev/dasda2` and `/dev/dasdb2` can be used to create a volume group. Click **Volume Management** as shown on Figure 16-5 on page 289.

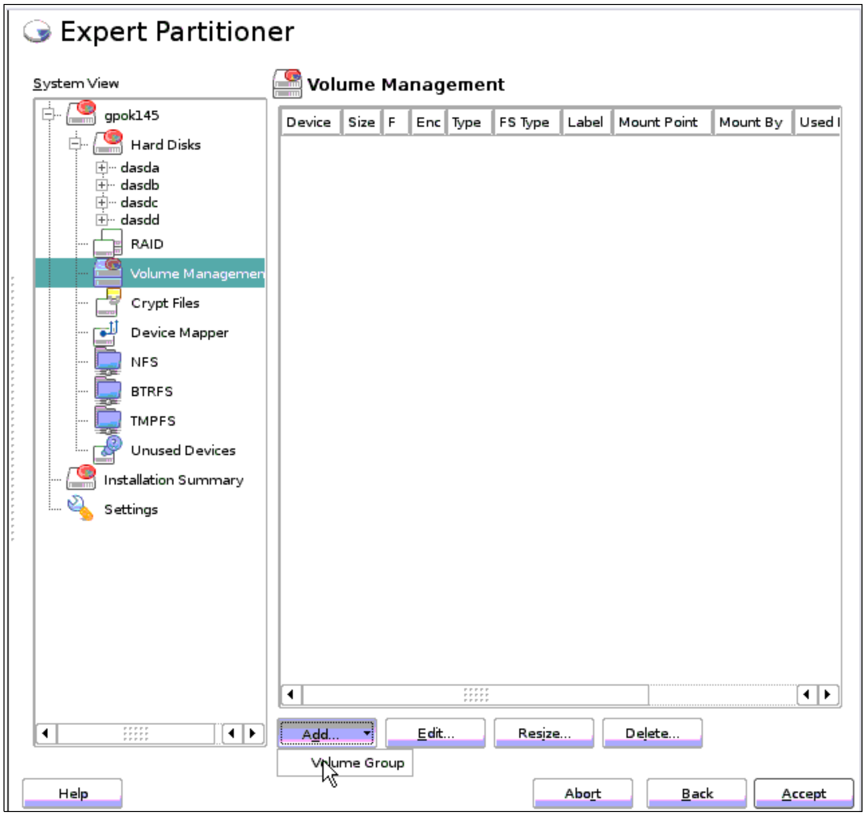


Figure 16-5 Creating logical volumes with volume management

22. Click **Add** => **Volume Group**.
23. At the *Add Volume Group* window, set the volume group name to **system_vg**.
24. Click **Add** after selecting both */dev/dasda2* and */dev/dasdb1* to the *Selected Physical Volumes,* as shown on Figure 16-6 on page 290. Click **Finish**.

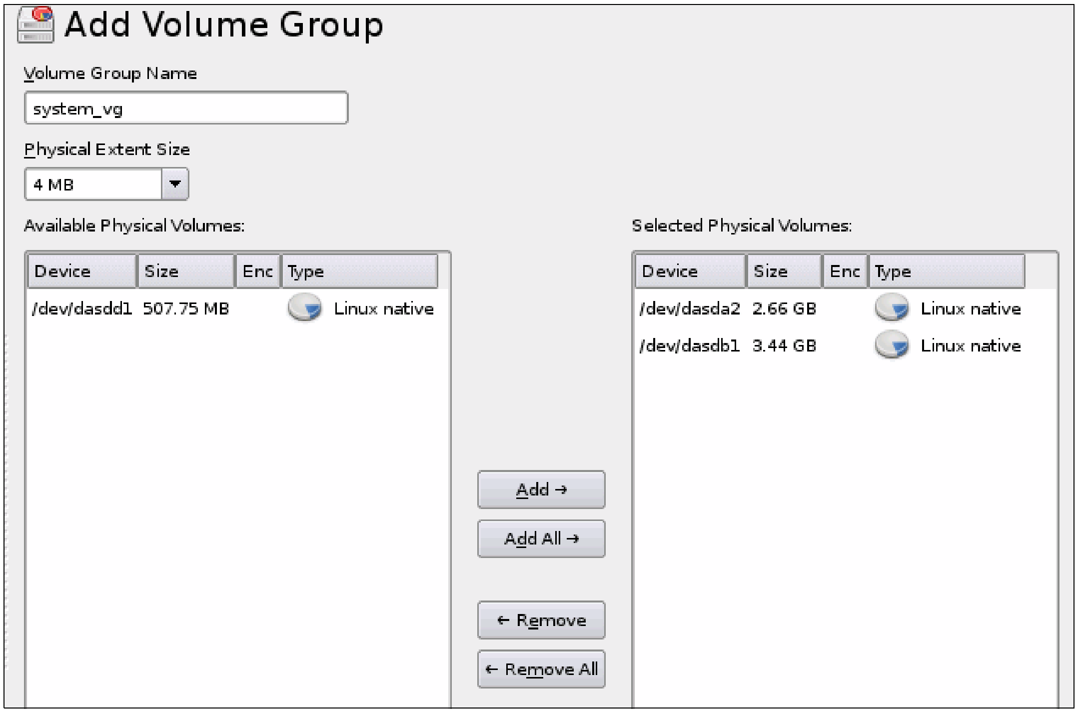


Figure 16-6 Add Volume Group panel

25. Back in *Expert Partitioner*, click the **plus sign** to the left of *Volume Management*. You should see the new **system-vg** volume group.
26. Click **system-vg** and the *Volume Group: /dev/system-vg* window will be displayed, as shown on Figure 16-7 on page 291. It shows there are no logical volumes defined. Click **Add**.

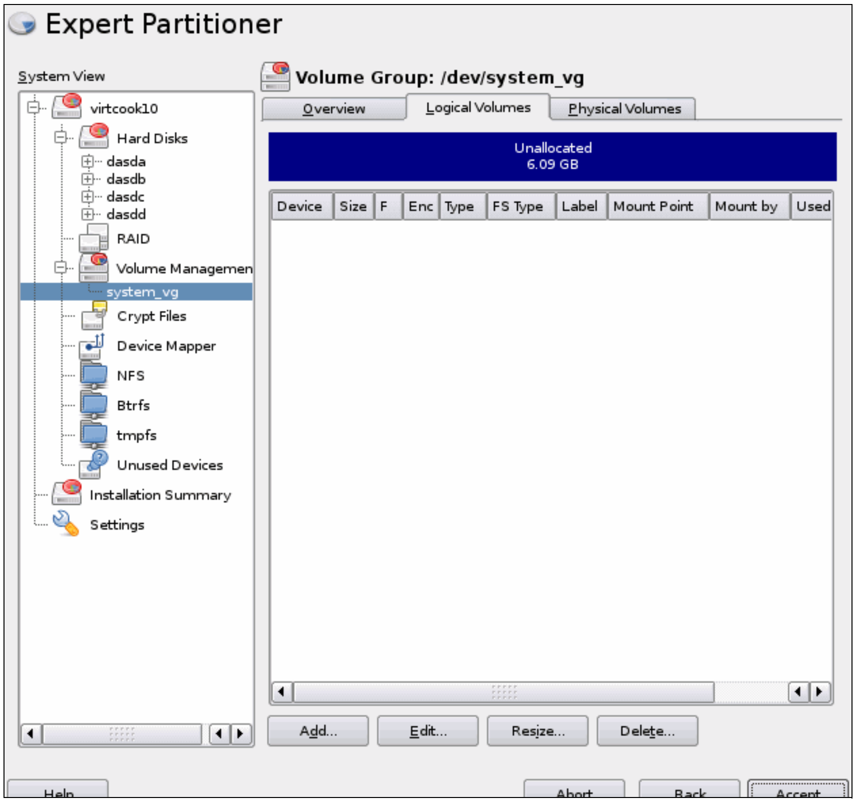


Figure 16-7   Volume Group view in Expert Partitioner panel

27.Enter **usr-lv** as the name for new logical volume and click **Next**.

28.On the *Add Logical Volume* window, click the *Custom Size* radio button and enter a size of **3 GB**. If you do not have HyperPAV available, accept the default of one stripe, else you can improve performance by setting this to two stripes. Click **Next.**

29.Accept the defaults of **ext3** as the *File System* type and the *Mount point* of **/usr**. Click **Finish**.

30.Create two more logical volumes similarly, using the information from Table 16-1.

Table 16-1   Logical volumes added to system-vg volume group

Mount point	**Logical volume name**	**Size**
`/usr/`	**usr-lv**	3 GB
`/var/`	**var-lv**	1 GB
`/srv/`	**srv-lv**	1 GB
`/tmp/`	**tmp-lv**	800 MB

You can choose other file systems and sizes. See 2.4, “Disk planning” on page 16 for more discussion. Because /srv/ often contains heavily used data, you can consider using **xfs** for that volume instead of **ext3**. The **xfs** file system can also be increased in size without interruption of service. If you use two stripes, you must also add two disks of same size to the volume group if you want to increase a logical volume.

31. Back in the *Expert Partitioner* window, click the top object in the *System View* (***gpok145*** in this example). You should see output similar to Figure 16-8. You do not have to format `/dev/dasdc1` (vdev `300`) and `/dev/dasdd1` (vdev `301`) because they are properly formatted as a Linux swap space by the **`SWAPGEN EXEC`**, and thus should be recognized as a swap space. Click **Accept**.

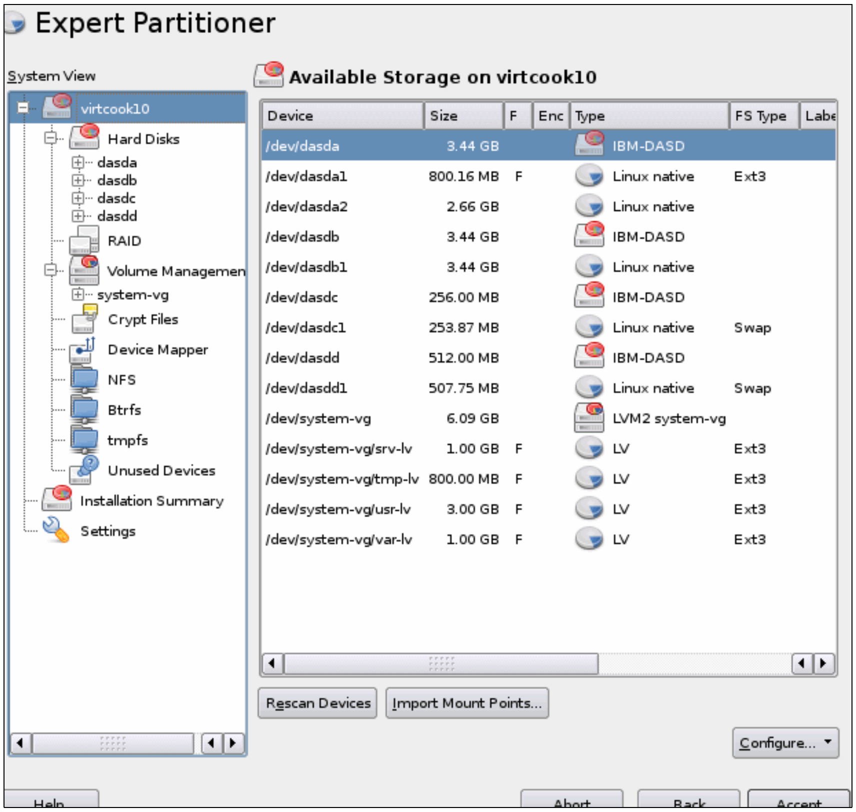


Figure 16-8 Disk partitioner: After customization

32. In the *Installation Settings* window, select **Software**. You might see a *Disk Space Warning* window. If so, click **OK**.

33. The *Software Selection* window opens. Leave **Base System, 32-Bit Runtime Environment**, **Help and Support Documentation**, and **Minimal System** selected. Clear all other setting as shown in Figure 16-9 on page 293**,** click **OK**. Click the disk usage window to update the graphs.

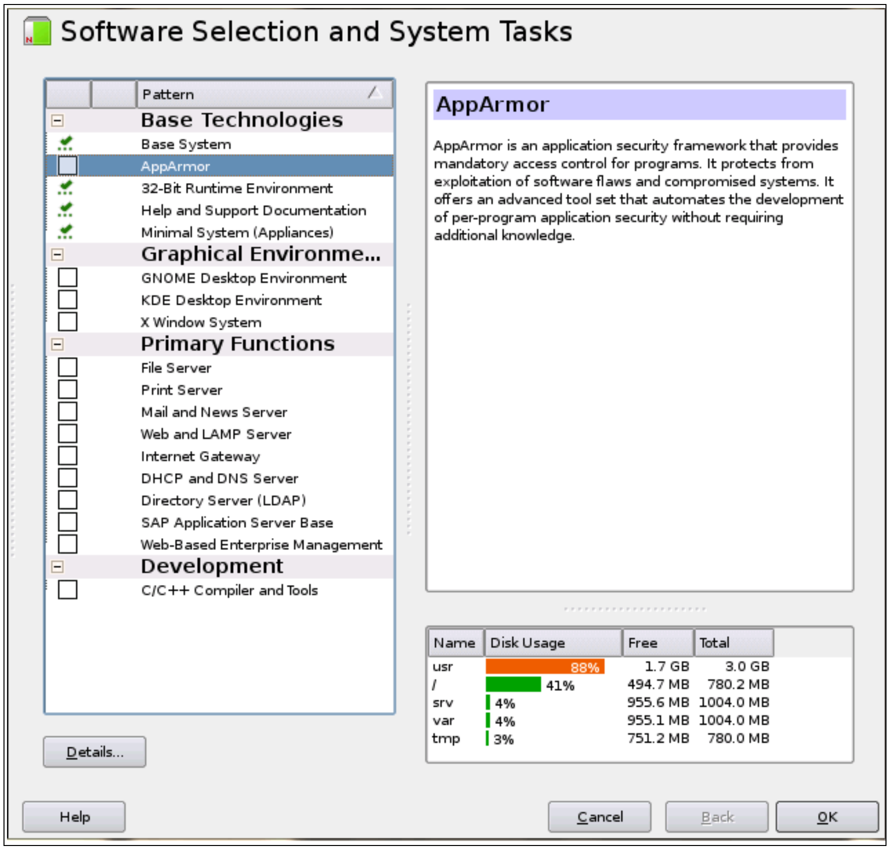


Figure 16-9 Software selection window

34. In the *Expert* tab, move down to *Default Runlevel* and choose **3: Full multiuser with network**, as shown in Figure 16-10. Click **OK**. You will receive a VNC warning. Click **Yes**.

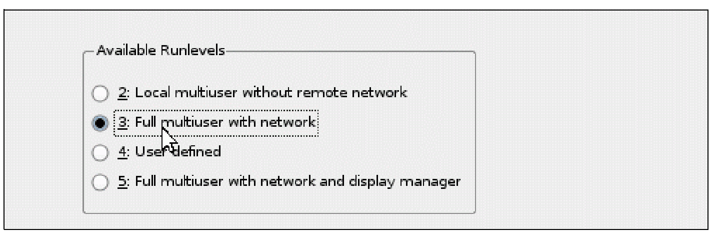


Figure 16-10 Default Runlevel

35. You are now ready to begin copying the RPMs onto your root file system. In the *Installation Settings* window, check the settings and click **Install**.
36. In the *Confirm Installation* window click **Install**.
37. The SUSE Linux Enterprise Server 11 SP3 system will be installed onto DASD. This should take about 3 - 10 minutes depending on a number of factors.
38. The window Finishing Basic Installation will appear, then the VNC client will end and the system will reboot. Go back to the 3270 console and clear the screen a number of times.
39. Restart or go back to the same VNC client used for the first part of installation. From a browser either click **Login Again** or click the browser's **refresh** button until another VNC login window appears. If you are using a VNC client, open that application again.
40. **Log in** using the same VNC password (**12345678** in this example).

41. In the Password for root user window, type the root password twice and click **Next**. *Do not forget* this password.

42. In the *Hostname and Domain Name* window, both *Hostname* and *Domain Name* are entered by the installer as they are specified in `S113GOLD PARM-S11` file. In this example, ***virtcook10*** is the host name and ***itso.ibm.com*** is the domain name. Clear the **Change Hostname via DHCP** check-box. Click **Next**.

43. In the *Network Configuration* window you will see *Firewall is enabled*, click the word **disable** to disable it, as shown in Figure 16-11. All other values should be correct; therefore, just click **Next**.

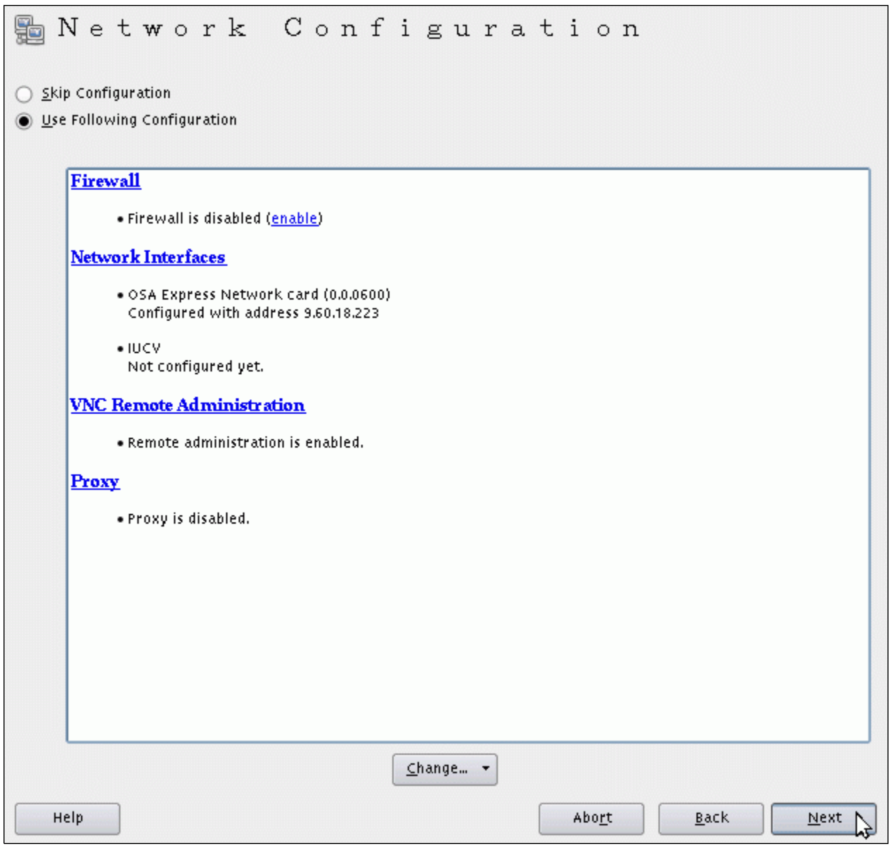


Figure 16-11 Disabling firewall in the Network Configuration window

44. In the *Test Internet Connection* window, if you do not have Internet access, select **No, skip this test**, and click **Next**.

45. In the *Network Services Configuration* window, accept the default and click **Next**. A certificate will be created.

46. In the *User Authentication Method* window, select **Local (/etc/passwd)** and click **Next**.

47. In the *New Local User* window, add at least one user to have a *non-root* ID on all cloned systems. When you are done, click **Next**.

48. In the *Writing the system configuration* window the **`SuSEconfig`** tool writes all your settings to disk.

49. The next window will be *Release Notes*. After reviewing the release notes, click **Next**.

50. In the *Hardware Configuration* window, choose the **Skip Configuration** radio button and click **Next**.

51. The last installation window is *Installation Completed*. Clear the box *Clone This System for AutoYaST2* check box and click **Finish.**

The VNC session should end. Return to the 3270 session and you may have to clear the screen a few times. Then, you should see a login prompt. You are done installing Linux! You can disconnect from the 3270 session using the **DISCONNECT** command:

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

You can now access this Linux system with SSH. If you have a Windows desktop, but do not have an SSH client configured, see 3.1, “PuTTY: A free SSH client for MS Windows” on page 36.

16.4 Configure the golden image

Customize the golden image before cloning. The following high-level steps are recommended.

1. “Configure the VNC server”
2. “Prepare for Online Update” on page 297
3. “Turn off unneeded services” on page 297
4. “Add the CMS file system RPM” on page 298
5. “Apply service: Online Update” on page 299
6. “Configure /etc/inittab” on page 299
7. “Disable cgroup memory” on page 300
8. “Configure SSH keys” on page 300
9. “Clean up temporary files” on page 303
10. “Reboot the system and verify changes” on page 303

16.4.1 Configure the VNC server

Applications often require a graphical environment. The `tightvnc` package is a Virtual Network Computing (VNC) server. It allows for a graphical environment to be set up easily using the **vncserver** command.

SUSE Linux Enterprise Server 11 SP3 configures a VNC connection, which starts by default. It is configured through the **xinetd** daemon. It is recommended that you disable it. VNC will be configured to run only when needed.

Perform the following steps:

1. Start an SSH session as **root** on the SUSE Linux Enterprise Server 11 SP3 golden image.
2. VNC is the only service that **xinetd** starts by default (this can be verified with the **chkconfig --list** command). You can turn off the **xinetd** service with the **service xinetd stop** command for this session and the **chkconfig** command at boot time:

```
# service xinetd stop
Shutting down xinetd:                                              done
# chkconfig xinetd off
```

Start the VNC server manually

When you first start the VNC server, you are prompted to set a password. After it is set, this will be the password that you will need to connect to it from a VNC client. Perform the following steps:

1. Start the VNC server:

```
# vncserver
You will require a password to access your desktops.

Password:
Verify:
Would you like to enter a view-only password (y/n)? n
xauth:  creating new authority file /root/.Xauthority

New 'X' desktop is virtcook10:1

Creating default startup script /root/.vnc/xstartup
Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/virtcook10:1.log
```

2. Stop the VNC server using the **-kill :1** argument:

```
# vncserver -kill :1
Killing Xvnc process ID 25338
```

3. The `icewm` package allows for the `IceWM`, a window manager that is more usable than the window manager (Tab Window Manager (**twm**)) that VNC uses by default. It is therefore recommended that you change to **icewm**. The package should be already installed on your system. Verify the RPM has been added:

```
# rpm -q icewm icewm-lite
icewm-1.2.36-1.35
icewm-lite-1.2.36-1.18
```

4. Change the window manager from **twm** to **icewm** in the file `/root/.vnc/xstartup`:

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

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

5. You might want to remove the `passwd` file so the cloned system does not have the same password as you just entered. If so, use the **rm passwd** command. If you remove the password file, when a system is cloned, the password will be prompted for the first time that the VNC server is initialized.

Verify the VNC server configuration

Perform the following steps:

1. Start the VNC server again:

```
# vncserver

New 'X' desktop is virtcook10:1

Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/virtcook10:1.log
```

2. Start a VNC client as shown on the left side of Figure 16-12. Enter the password and the resulting VNC session is shown on the right side of Figure 16-12.

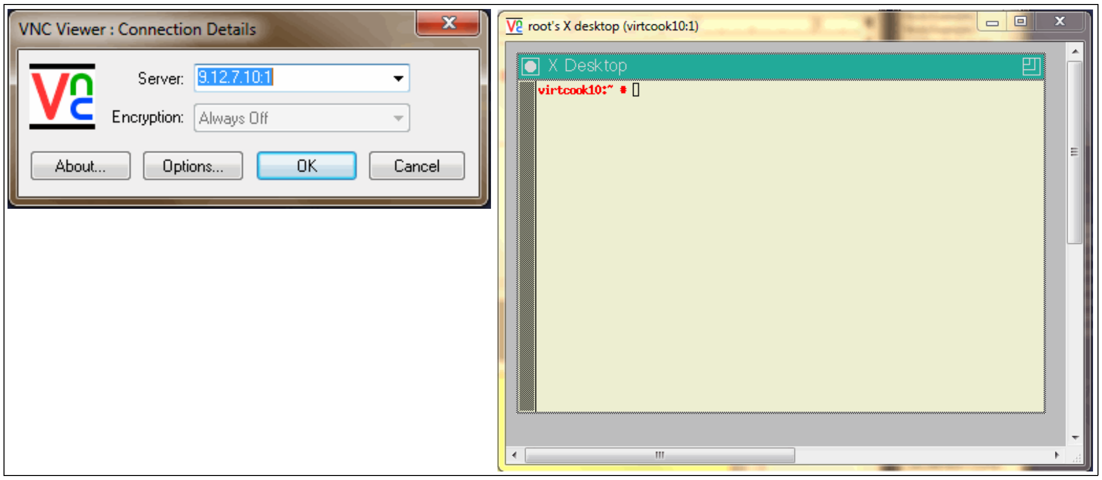


Figure 16-12 VNC session

   The VNC server will not be started across reboots. When you need a graphical environment, start the **vncserver** manually.

3. Stop the VNC server using the **-kill:1** argument:

```
# vncserver -kill :1
Killing Xvnc process ID 25398
```

The VNC server is now configured to be started upon request.

16.4.2 Prepare for Online Update

This step is recommended if you have Internet access.

> **Note:** The system used for the writing of this book did not have Internet access, so the following steps have not been tested for SUSE Linux Enterprise Server 11 SP3.

Before you can use *Online Update* (recommended), online sources have to be configured. This is done through the Novell Customer Center Configuration. To configure the Customer Center, a web browser is needed. For this reason, a VNC server session must be started.

A graphical environment is recommended for this step. Start a VNC viewer. In a terminal session, start YaST with command **yast2** and choose the **Software => Online Update Configuration** selection.

16.4.3 Turn off unneeded services

There are a number of services that are started in a SUSE Linux Enterprise Server 11 SP3 minimal system. Perform the following steps:

1. View the services that are on in run level 3 with the following **chkconfig** command:

```
# chkconfig -l | grep 3:on
auditd                    0:off  1:off  2:off  3:on   4:off  5:on   6:off
cron                      0:off  1:off  2:on   3:on   4:off  5:on   6:off
dbus                      0:off  1:off  2:on   3:on   4:off  5:on   6:off
```

```
earlysyslog               0:off  1:off  2:on   3:on   4:off  5:on   6:off
fbset                     0:off  1:on   2:on   3:on   4:off  5:on   6:off
haldaemon                 0:off  1:off  2:on   3:on   4:off  5:on   6:off
haveged                   0:off  1:off  2:on   3:on   4:off  5:on   6:off
irq_balancer              0:off  1:on   2:on   3:on   4:off  5:on   6:off
network                   0:off  1:off  2:on   3:on   4:off  5:on   6:off
network-remotefs          0:off  1:off  2:on   3:on   4:off  5:on   6:off
nfs                       0:off  1:off  2:off  3:on   4:off  5:on   6:off
nscd                      0:off  1:off  2:off  3:on   4:off  5:on   6:off
postfix                   0:off  1:off  2:off  3:on   4:off  5:on   6:off
purge-kernels             0:off  1:off  2:off  3:on   4:off  5:on   6:off
random                    0:off  1:off  2:on   3:on   4:off  5:on   6:off
rpcbind                   0:off  1:off  2:off  3:on   4:off  5:on   6:off
smartd                    0:off  1:off  2:on   3:on   4:off  5:on   6:off
splash                    0:off  1:on   2:on   3:on   4:off  5:on   6:off  S:on
splash_early              0:off  1:off  2:on   3:on   4:off  5:on   6:off
sshd                      0:off  1:off  2:off  3:on   4:off  5:on   6:off
syslog                    0:off  1:off  2:on   3:on   4:off  5:on   6:off
```

2. In order to keep the golden image as lean as possible in terms of CPU usage, some of these services can be turned off. Turn off the following services by using the following **chkconfig** commands:

```
# chkconfig auditd off
# chkconfig fbset off
# chkconfig network-remotefs off
# chkconfig postfix off
# chkconfig splash off
# chkconfig splash_early off
# chkconfig smartd off
```

3. You can choose to leave these services on, or turn off others. You can review which services are now configured to start in run level 3 with the following command:

```
# chkconfig -l | grep 3:on
cron                      0:off  1:off  2:on   3:on   4:off  5:on   6:off
dbus                      0:off  1:off  2:on   3:on   4:off  5:on   6:off
earlysyslog               0:off  1:off  2:on   3:on   4:off  5:on   6:off
haldaemon                 0:off  1:off  2:on   3:on   4:off  5:on   6:off
haveged                   0:off  1:off  2:on   3:on   4:off  5:on   6:off
irq_balancer              0:off  1:on   2:on   3:on   4:off  5:on   6:off
network                   0:off  1:off  2:on   3:on   4:off  5:on   6:off
nfs                       0:off  1:off  2:off  3:on   4:off  5:on   6:off
nscd                      0:off  1:off  2:off  3:on   4:off  5:on   6:off
purge-kernels             0:off  1:off  2:off  3:on   4:off  5:on   6:off
random                    0:off  1:off  2:on   3:on   4:off  5:on   6:off
rpcbind                   0:off  1:off  2:off  3:on   4:off  5:on   6:off
sshd                      0:off  1:off  2:off  3:on   4:off  5:on   6:off
syslog                    0:off  1:off  2:on   3:on   4:off  5:on   6:off
```

This shows the services that will run in the default run level of 3.

16.4.4 Add the CMS file system RPM

To read CMS files from Linux, the cmsfs RPM is used. The clone script requires it to read parameter files. To add this package to the golden image, use:

```
# zypper install cmsfs
Loading repository data...
Reading installed packages...
Resolving package dependencies...
```

```
The following NEW package is going to be installed:
  cmsfs

1 new package to install.
Overall download size: 34.0 KiB. After the operation, additional 148.0 KiB will
be used.
Continue? [y/n/?] (y): y
Retrieving package cmsfs-1.1.8-6.4.1.s390x (1/1), 34.0 KiB (148.0 KiB unpacked)
Installing: cmsfs-1.1.8-6.4.1 [done]
```

16.4.5 Apply service: Online Update

If you have a SUSE Maintenance web account, you can use it to retrieve the latest patches for SUSE Linux Enterprise Server 11 SP3. Because many of these patches contain security and bug fixes, it is recommended that you apply the patches for the golden image so that it is up to date. Subsequently all the servers you clone after the golden image will also be up to date.

Section 16.4.2, “Prepare for Online Update” on page 297 must be completed before proceeding.

The system used for the writing of this book did not have Internet access, so this part was not tested at all. You can choose which packages to install and which not to install. It is highly recommended to download security patches.

When you finish here, do not reboot yet. You will reboot shortly after boot configuration changes are made.

16.4.6 Configure /etc/inittab

If you did not change the default runlevel from 5 to 3 during the installation process, this would be a good time to do so. By default, SUSE Linux Enterprise Server 11 boots into run level 5, which is designed for a graphical environment. To keep the golden image as lean as possible, it is recommended that this be reset to run level 3. This is set in the `/etc/inittab` file with the variable `initdefault`.

Linux reboots when a `Ctrl-Alt-Del` key sequence is trapped. This key sequence is simulated by z/VM when it issues a **SIGNAL SHUTDOWN** command. Rather than rebooting, it might be better that your system halts (shuts down).

Change **shutdown -r** to **shutdown -h**:

```
# cd /etc
# cp inittab inittab.orig
# vi inittab
...
# The default runlevel is defined here
id:3:initdefault:
...
# what to do when CTRL-ALT-DEL is pressed
ca::ctrlaltdel:/sbin/shutdown -h -t 4 now
...
```

This change will be picked up when the system is rebooted.

16.4.7 Disable cgroup memory

Linux Control Groups, or *cgroups* are collections of processes with the same criteria. They enable you to limit, account, and isolate resource usage. They are not commonly used on System z Linux. Up to 1% of the memory on each Linux system can be reclaimed by disabling cgroup memory.

Unless you know why you need this feature, the following steps are recommended to disable it:

1. Make a backup copy of the `/etc/zipl.conf` file:

```
# cd /etc
# cp zipl.conf zipl.conf.orig
```

2. Add the kernel parameter `cgroup_disable=memory`:

```
# vi zipl.conf
# Modified by YaST2. Last modification on Wed Jul 17 16:35:24 EDT 2013
[defaultboot]
defaultmenu = menu

###Don't change this comment - YaST2 identifier: Original name: linux###
[SLES11_SP3]
    image = /boot/image-3.0.76-0.9-default
    target = /boot/zipl
    ramdisk = /boot/initrd-3.0.76-0.9-default,0x2000000
    parameters = "root=/dev/disk/by-path/ccw-0.0.0100-part1 cgroup_disable=memory
hvc_iucv=8 TERM=dumb resume=/dev/disk/by-path/ccw-0.0.0300-part1"
...
```

Note: The `parameters=` line is shown on two lines for clarity. It is really just one line.

3. Run the **zipl** command for the change to take effect:

```
# zipl
Using config file '/etc/zipl.conf'
Building bootmap in '/boot/zipl'
Building menu 'menu'
Adding #1: IPL section 'SLES11_SP3' (default)
Adding #2: IPL section 'Failsafe_1'
Adding #3: IPL section 'ipl'
Preparing boot device: dasda (0100).
Done.
```

16.4.8 Configure SSH keys

SSH sessions are typically authenticated using passwords typed in from the keyboard. With SSH *key-based authentication* sessions can be authenticated using public and private keys so that no password is needed. To accomplish this, the following statements must be true:

- The SSH server system must have the client's public key.
- The SSH client must send its private key.
- The keys must match cryptographically.

SSH key-based authentication can be set up from the Linux administration system (client) to the virtual servers. If the preceding requirements are met, then key-based authentication will work to the cloned virtual servers.

To set up key-based authentication from the Linux administration system to the golden image (and consequently from Linux administration system to all cloned images), perform the steps that follow. You will need SSH sessions on both the SUSE Linux Enterprise Server 11 SP3 golden image (***virtcook10***) and the SLES 11 SP3 Linux administration system (LNXADMIN), so prefixes are used in the following examples:

1. On the golden image, see if the directory /root/.ssh/ exists:

```
virtcook10: # cd /root
virtcook10: # ls -d .ssh
ls: cannot access .ssh: No such file or directory
```

2. In this example, it does not exist. Create the directory and set the permission bits to octal 700 with the **chmod** command:

```
virtcook10: # mkdir .ssh
virtcook10: # chmod 700 .ssh
```

3. **Start an SSH session as root** to the Linux administration system. In this example, the IP address is ***9.12.7.8***.

4. Create a new key of type DSA in root's .ssh/ directory with the **ssh-keygen** command. Accept all defaults:

```
virtcook8:~ # cd .ssh
virtcook8:~/.ssh # ssh-keygen -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/root/.ssh/id_dsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_dsa.
Your public key has been saved in /root/.ssh/id_dsa.pub.
The key fingerprint is:
e5:1d:dc:4b:06:15:e1:f5:43:89:34:5d:b7:12:8c:65 [MD5] root@linuxadmin
The key's randomart image is:
+--[ DSA 1024]----+
|            *E*+*|
|           o.*+++|
|          . o.=o.|
|         o . +...|
|        S . . .  |
|                 |
|                 |
|                 |
|                 |
+--[MD5]----------+
```

5. Show the newly created files:

```
# ls -l
total 12
-rw------- 1 root root 672 Jul 18 08:35 id_dsa
-rw-r--r-- 1 root root 604 Jul 18 08:35 id_dsa.pub
-rw-r--r-- 1 root root 392 Jul 15 15:54 known_hosts
```

   This shows that the id_dsa (private key) and id_dsa.pub (public key) were created.

6. Copy the new public key to the /root/.ssh/ directory that you just created to the golden image (***9.12.7.10*** in this example) as the file /root/.ssh/authorized_keys:

```
virtcook8:~/.ssh # scp id_dsa.pub 9.12.7.10:/root/.ssh/authorized_keys
The authenticity of host '9.12.7.10 (9.12.7.10)' can't be established.
ECDSA key fingerprint is 81:19:f0:20:d4:c3:22:26:eb:ba:4f:19:52:10:87:8e [MD5].
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '9.12.7.10' (ECDSA) to the list of known hosts.
Password:
id_dsa.pub                                        100%  604     0.6KB/s   00:00
```

The `authorized_keys` file sets which keys are authorized to get access without requiring a password.

7. You should now be able to get an SSH session to the golden image without needing a password because the Linux administration system is recognized as an authorized host:

```
virtcook8: # ssh 9.12.7.10
Last login: Fri Jun  7 13:54:08 2013 from gunreben.itso.ibm.com
virtcook10:~ # exit
logout
Connection to 9.12.7.10 closed.
```

If you get an SSH session without having to supply a password, as with the preceding example, it shows that key-based authentication is working from the Linux administration system to the golden image. All systems cloned from this image will also allow such access.

16.4.9 Copy set up the golden image for initial boot

A new method of cloning has been adopted in this book: A boot script is copied to this golden image. This script retrieves its network configuration from the CMS parameter file, which can also be used to install Linux manually.

To enable this feature, perform the following steps:

1. Log on to the golden image as user `root`.
2. Change directory to `/etc/init.d/`:

```
# cd /etc/init.d
```

3. Copy the file `boot.config` from the Linux administration server to the current directory:

```
# scp virtcook8:/srv/ftp/SG248147/sles11sp3/boot.clone .
The authenticity of host 'virtcook8 (9.12.7.8)' can't be established.
ECDSA key fingerprint is 01:cf:f6:03:96:c6:86:68:4a:7e:a4:a3:33:59:ea:e3 [MD5].
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'virtcook8,9.12.7.8' (ECDSA) to the list of known hosts.
Password:
boot.clone                                        100% 6262     6.1KB/s   00:00
```

4. Enable the script to run during bootup with the **chkconfig** command:

```
# chkconfig boot.clone on
```

5. Verify that the script has been inserted into the boot run level:

```
# ls boot.d
K01boot.cleanup     K02boot.clock        S09boot.lvm
K01boot.clone       K02boot.localnet     S10boot.localfs
K01boot.compliance  K02boot.swap         S12boot.compliance
K01boot.cpi         K04boot.localfs      S13boot.clone
K01boot.debugfs     K05boot.lvm          S13boot.klog
...
```

The golden image is now ready to replace its current network configuration with the values found in a CMS file `<USERID> PARM-S11` on the 191 disk.

16.4.10 Configure boot.local

In order to be able to use the live guest relocation (LGR) feature of z/VM, there must not be any links to CMS. In z/VM the default is to have several disks available, like the CMS disk or the 191 disk. In the setup described in this book, `LNXDFLT PROFILE` creates links to minidisks and assigns the virtual device numbers 190, 191, 19D, and 19E. To check with your local installation, use the following command:

```
# modprobe vmcp
# vmcp q v dasd | grep 019
DASD 0190 3390 JV1130 R/O       214 CYL ON DASD  1130 SUBCHANNEL = 0006
DASD 0191 3390 JM1262 R/O       500 CYL ON DASD  1262 SUBCHANNEL = 0009
DASD 019D 3390 JV1130 R/O       292 CYL ON DASD  1130 SUBCHANNEL = 0007
DASD 019E 3390 JV1130 R/O       500 CYL ON DASD  1130 SUBCHANNEL = 0008
```

This shows that there are four CMS disks. All of these must be detached before LGR. One way to accomplish this is to add commands to the file /etc/init.d/boot.local - a script that is run after the initial boot process and before switching to the starting run level.

In addition to detaching CMS disks, the **chshut** command is also added twice so that the virtual machine is logged off after it is halted or powered off. This will allow for an orderly shutdown of z/VM.

To do so, add the following four lines to boot.local:

```
# vi /etc/init.d/boot.local
...
# Here you should add things, that should happen directly after booting
# before we're going to the first run level.
#
modprobe vmcp
vmcp det 190 191 19d 19e
chshut halt vmcmd logoff
chshut poff vmcmd logoff
```

16.4.11 Clean up temporary files

Now is the best time to clean temporary files from golden image. If they stay there, they will be copied to each clone and occupy space there. There are a number of directories that can be cleaned up and the history of the commands that are executed can be cleared.

1. Create a script **cloneprep.sh** that can be run to prepare for cloning:

```
# cd /usr/local/sbin
# vi cloneprep.sh
#!/bin/bash
rm -fr /tmp/*
rm -fr /var/log/YaST2/*
history -c
```

   You can choose different steps to prepare for cloning.

2. Make the script executable with the **chmod +x** command:

```
# chmod +x cloneprep.sh
```

3. Call the script interactively:

```
# cloneprep.sh
```

The system should now be cleaned up for cloning.

16.4.12 Reboot the system and verify changes

You are now done customizing the golden Linux image. Now **reboot** to test your changes:

```
# reboot
```

```
Broadcast message from root (pts/0) (Mon Nov 30 08:51:49 2009):

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

When the system comes back up, you should verify the changes that you made:

1. **SSH back into the golden image** and check a few settings.
2. Use the **df -h** command to display your file systems:

```
# df -h
Filesystem                     Size  Used Avail Use% Mounted on
/dev/dasda1                    788M  165M  583M  23% /
udev                           498M  136K  498M   1% /dev
tmpfs                          498M     0  498M   0% /dev/shm
/dev/mapper/system--vg-srv--lv 1008M   34M  924M   4% /srv
/dev/mapper/system--vg-tmp--lv  788M   17M  731M   3% /tmp
/dev/mapper/system--vg-usr--lv  3.0G  1.3G  1.6G  44% /usr
/dev/mapper/system--vg-var--lv 1008M   87M  871M   9% /var
```

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

```
# swapon -s
Filename                                Type            Size    Used    Priority
/dev/dasdc1                             partition       259956  0       -1
/dev/dasdd1                             partition       519924  0       -2
```

   Device with higher priority will be used first.

4. The script **boot.clone** disables itself after bootup. Reenable it with the following **chkconfig** command:

```
# chkconfig boot.clone on
```

5. Shut down your golden image from the SSH session (to clone Linux, it should be shut down):

```
# shutdown -h now
```

Congratulations. You have now successfully installed the golden image. This image should normally be shut down because it is not recommended to clone a running Linux system.

17

Clone SUSE Linux Enterprise Server 11 SP3

"*The whole of science is nothing more than a refinement of everyday thinking.*"

— Albert Einstein

This chapter focuses on cloning the SUSE Linux Enterprise Server 11 SP3 golden image. Examples are given for cloning manually and by using a script. The following steps are described:

1. "Define three new virtual machines"
2. "Clone a virtual server manually" on page 306
3. "Clone a virtual server automatically" on page 310

17.1 Define three new virtual machines

In this section, you define three new virtual machines that will be targets to clone to. In this example, they are named `LINUX4`, `LINUX5`, and `LINUX6` (each user ID suffix corresponds to the last dotted decimal octet in the associated IP address).

To do so, perform the following steps:

1. Log on to `MAINT`.
2. Edit the `USER DIRECT` file to add new virtual machines for Linux:

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

3. Go to the bottom of the file and add the following lines. In this example, the user IDs will be `LINUX4`, `LINUX5`, and `LINUX6` with a password of `LNX4VM`. They will default to have 512 MB of memory but can be set up to 1 GB. They will have only G privilege class (General user). They will each have two 5008 cylinder minidisks. In this example, the three 3390-9s assigned have labels of ***JM1269***, ***JM126A***, and ***JM126B***.

These disks should have been formatted in section 5.8.4, “Format DASD for minidisks” on page 84. Replace the labels with those being used in your organization:

```
*
USER LINUX4 LNX4VM 512M 1G G
 INCLUDE LNXDFLT
 MDISK 100 3390 0001 5008 JM1269 MR LNX4VM LNX4VM LNX4VM
 MDISK 101 3390 5009 5008 JM1269 MR LNX4VM LNX4VM LNX4VM
*
USER LINUX5 LNX4VM 512M 1G G
 INCLUDE LNXDFLT
 MDISK 100 3390 0001 5008 JM126A MR LNX4VM LNX4VM LNX4VM
 MDISK 101 3390 5009 5008 JM126A MR LNX4VM LNX4VM LNX4VM
*
USER LINUX6 LNX4VM 512M 1G G
 INCLUDE LNXDFLT
 MDISK 100 3390 0001 5008 JM126B MR LNX4VM LNX4VM LNX4VM
 MDISK 101 3390 5009 5008 JM126B MR LNX4VM LNX4VM LNX4VM
```

4. You might need to add the new volumes to the `$ALLOC$` virtual machine so cylinder 0 will not show up in the disk map as a gap.
5. Again check for gaps and overlaps:

```
==> diskmap user (doends
==> x user diskmap
====> all /gap/|/overlap/
...
```

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

```
==> directxa user
z/VM USER DIRECTORY CREATION PROGRAM - VERSION 6 RELEASE 3.0
EOJ DIRECTORY UPDATED AND ON LINE
HCPDIR494I User directory occupies 109 disk pages
Ready; T=0.01/0.02 07:58:39
```

Three new virtual machines should now exist.

17.2 Clone a virtual server manually

Before using the shell script **clone.sh** to clone a server, you might want to clone a server manually to better understand the process. There are many ways to clone Linux under z/VM. The steps in this section are just one way to do it. The following assumptions are made based on what you have done so far:

- The source (golden) Linux image is on the virtual machine `S113GOLD` on minidisks `100` and `101`.
- The target virtual machine, `LINUX4` in this example, has identically sized minidisks.
- The **vmcp** command is available to the Linux administration system, `LNXADMIN`, to issue z/VM CP commands.
- The z/VM **FLASHCOPY** command can be used but if you do not have that support, the Linux **dasdfmt** and **dd** commands can also be used.

Given these assumptions, one set of steps that can be used to clone a system is as follows:

1. Link the source disks read-only.
2. Link the target disks read/write.

3. Copy the source to the target disk using **FLASHCOPY** or the Linux **dasdfmt** and **dd** commands.
4. Mount the newly copied root file system.
5. Modify the networking information about the target system.
6. Detach the target disks.
7. IPL the target system.
8. Modify the SSH keys on the target system.

The following sections describe these steps in detail.

Link the source disks read-only

Start an SSH session as root to the Linux administration system, LNXADMIN.

The source minidisks at virtual addresses 100 and 101 are on the golden image, S113GOLD. They are linked read-only as virtual addresses 1100 and 1101 using the RR parameter to the **CP LINK** command:

```
# vmcp link s113gold 100 1100 rr
# vmcp link s113gold 101 1101 rr
```

Link the target disk read/write

The target minidisks, also at addresses 100 and 101 are on the Linux administration system, LINUX153. They are linked multi-read (read/write if no other virtual machine has write access) using the MR parameter as virtual device 2100 and 2101 using the **CP LINK** command:

```
# vmcp link linux4 100 2100 mr
# vmcp link linux4 101 2101 mr
```

Copy the source to the target disk using FLASHCOPY

The two disks are copied using the **CP FLASHCOPY** command:

```
# vmcp flashcopy 1100 0 end to 2100 0 end
Command complete: FLASHCOPY 1100 0 5007 TO 2100 0 5007
# vmcp flashcopy 1101 0 end to 2101 0 end
Command complete: FLASHCOPY 1101 0 5007 TO 2101 0 5007
```

Note: If you do not have **FLASHCOPY** support, you can use the Linux **dasdfmt** and **dd** commands. You must first enable the source and target disks using the **chccwdev -e** command, then determine the device name using the **lsdasd** command:

```
# chccwdev -e 1100
Setting device 0.0.1100 online
Done
# chccwdev -e 1101
Setting device 0.0.1101 online
Done
# chccwdev -e 2100
Setting device 0.0.2100 online
Done
# chccwdev -e 2101
Setting device 0.0.2101 online
Done
# lsdasd
# lsdasd
Bus-ID     Status      Name      Device  Type  BlkSz  Size      Blocks
==============================================================================
0.0.0100   active      dasda     94:0    ECKD  4096   2347MB    600840
0.0.0300   active      dasdb     94:4    FBA   512    256MB     524288
0.0.0301   active      dasdc     94:8    FBA   512    512MB     1048576
0.0.1100   active      dasdd     94:12   ECKD  4096   2347MB    600840
0.0.1101   active      dasde     94:16   ECKD  4096   2347MB    600840
0.0.2100   active      dasdf     94:20   ECKD  4096   2347MB    600840
0.0.2101   active      dasdg     94:24   ECKD  4096   2347MB    600840
```

In this example, the source devices are /dev/dasdd and /dev/dasde, and the target devices are /dev/dasdf and /dev/dasdg. Format the target disks using the **dasdfmt** command, then copy it using the **dd** command using a block size of 4 K (4096) bytes:

```
# dasdfmt -b 4096 -y -f /dev/dasdf
...
# dasdfmt -b 4096 -y -f /dev/dasdg
...
# dd if=/dev/dasdd of=/dev/dasdf bs=4096
600840+0 records in
600840+0 records out
# dd if=/dev/dasde of=/dev/dasdg bs=4096
600840+0 records in
600840+0 records out
```

The golden image should now be copied to the target disks. Disable the disks:

```
# chccwdev -d 1100
# chccwdev -d 1101
# chccwdev -d 2100
# chccwdev -d 2101
```

Change directory to /, and use the **sync** command to flush the disks:

```
# cd /
# sync
```

Detach the target disks

Change directory to /, use the **sync** command to flush the disks:

```
# cd /
# sync
```

Detach the source and target minidisks using the **CP DETACH** command:

```
# vmcp det 1100 1101 2100 2101
1100 1101 2100 2101 DETACHED
```

The newly copied and modified system disks have now been detached.

Create a parameter file for the target system

The network information is gathered from a CMS file at boot time. The **boot.clone** script in the golden image expects a parameter file named <USERID> PARM-S11 on its 191 disk.

In this example, the file will be LINUX4 PARM-S11. To create this file on the LNXMAINT 192 disk, perform the following steps:

1. Log on to LNXMAINT.
2. Copy the S113GOLD PARM-S11 file to the target file - in this example the file name is LINUX4:

```
==> copy s113gold parm-s11 d linux4 = =
```

3. Edit the file and change the IP address and host name to values for your organization:

```
==> x linux4 parm-s11 d
ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dumb
HostIP=9.12.7.4         Hostname=virtcook4.itso.ibm.com
...
```

4. Save the file and logoff of LNXMAINT.

The **boot.clone** script now can obtain the correct network information for the new clone. A side effect of this model is that the same information could be used for a manual Linux installation if necessary.

IPL the target system

You should now be ready to IPL the manually cloned system:

1. Start a 3270 session to the new clone. In this example, it is ***LINUX4***.
2. IPL from minidisk 100:

```
00: z/VM Version 6 Release 3.0, Service Level 0000 (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 15:16:43 EDT FRIDAY 06/07/13
00: Command complete
00: NIC 0600 is created; devices 0600-0602 defined
00: NIC 0600 is connected to VSWITCH SYSTEM VSW1
z/VM V6.3.0    2012-08-21 20:25

DMSACP723I A (191) R/O
DMSACP723I C (592) R/O
DIAG swap disk defined at virtual address 300 (64988 4K pages of swap space)
DIAG swap disk defined at virtual address 301 (129980 4K pages of swap space)
Do you want to IPL Linux from minidisk 100? y/n
y
00: zIPL v1.15.0-0.136.3 interactive boot menu
00:
00:  0. default (SLES11_SP3)
00:
00:  1. SLES11_SP3
00:  2. Failsafe_1
00:  3. ipl
```

```
00:
00: Note: VM users please use '#cp vi vmsg <number> <kernel-parameters>'
00:
00: Please choose (default will boot in 10 seconds):
00: Booting default (SLES11_SP3)...
...
```

Watch for error messages on the console. Your new system should come up cleanly using the modified IP address and host name.

Congratulations. You have now cloned a Linux system manually. You can look around the new system. It should be identical to the golden image except for the IP address and host name.

Next, you can learn how to do it automatically. You will use the `LINUX4` virtual machine again. To clone, the target virtual machine must be logged off. You could shut down the new system cleanly, but because you will be cloning again, it does not matter. Go to the 3270 session and you can crash the `LINUX4` virtual machine by using the **`LOGOFF`** command:

```
==> #cp log
```

17.3 Clone 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.

To clone a virtual server automatically, perform the following steps:

1. Start an SSH session as `root` to the Linux administration system.
2. Copy the **`clone.sh`** script from the unpacked tar file to `/usr/local/sbin/`:

   ```
   # cp /srv/ftp/SG248147/sles11sp3/clone.sh /usr/local/sbin
   ```

3. Clone the golden image to the target virtual machine, `LINUX4` in this example. Both the source and target virtual machines must be logged off. The script reads the parameter file on the `LNXMAINT 192` disk (the Linux administration system's `191` disk) to obtain information necessary to give the new Linux virtual server an identity. It should take less than a minute to clone with **`FLASHCOPY`** support and 5-20 minutes without it.

   The following example shows cloning to the `LINUX4` virtual machine with **`FLASHCOPY`** support. The output is divided into sections:

   ```
   # clone.sh from s113gold to linux4
   Checking that S113GOLD exists and is not logged on ...
   Invoking CP command: QUERY S113GOLD
   HCPCQU045E S113GOLD not logged on
   Error: non-zero CP response for command 'QUERY S113GOLD': #45
   Checking that LINUX4 exists and is not logged on ...
   Invoking CP command: QUERY LINUX4
   HCPCQU045E LINUX4 not logged on
   Error: non-zero CP response for command 'QUERY LINUX4': #45

   WARNING!!: Minidisks 100 and 101 will be copied to LINUX4
   Network data is retrieved from LINUX4 PARM-S11 on 191 disk
   during the first boot of LINUX4
   Are you sure you want to overwrite these disks (y/n): y
   ```

The script verifies that the virtual machines `S113GOLD` and `LINUX4` exist and are logged off. The `191` disk is deactivated and reactivated to pick up any changes that might have occurred. You are then asked to confirm that the disks on the target system can be written to. Answer yes with **y**.

The script continues:

```
Linking source and target 100 disks ...
Invoking CP command: detach 1100
HCPDTV040E Device 1100 does not exist
Error: non-zero CP response for command 'DETACH 1100': #40
Invoking CP command: link S113GOLD 100 1100 rr

Invoking CP command: detach 2100
HCPDTV040E Device 2100 does not exist
Error: non-zero CP response for command 'DETACH 2100': #40
Invoking CP command: link LINUX4 100 2100 mr

Copying 100 disks ...

FLASHCOPYing 1100 to 2100 ...
Invoking CP command: FLASHCOPY 1100 0 end to 2100 0 end
Command complete: FLASHCOPY 1100 0 5007 TO 2100 0 5007
Take 1100 Offline....
Device is already offline
Done
Invoking CP command: det 1100
DASD 1100 DETACHED
Invoking CP command: det 2100
DASD 2100 DETACHED

---------------------------------------
Linking source and target 101 disks ...
Invoking CP command: detach 1101
HCPDTV040E Device 1101 does not exist
Error: non-zero CP response for command 'DETACH 1101': #40
Invoking CP command: link S113GOLD 101 1101 rr

Invoking CP command: detach 2101
HCPDTV040E Device 2101 does not exist
Error: non-zero CP response for command 'DETACH 2101': #40
Invoking CP command: link LINUX4 101 2101 mr

Copying 101 disks ...

FLASHCOPYing 1101 to 2101 ...
Invoking CP command: FLASHCOPY 1101 0 end to 2101 0 end
Command complete: FLASHCOPY 1101 0 5007 TO 2101 0 5007
Taking 1101 Offline...
Device is already offline
Done
Invoking CP command: det 1101
DASD 1101 DETACHED
Taking 2101 Offline...
Device is already offline
Done
Invoking CP command: det 2101
DASD 2101 DETACHED
```

In the section above, the script copies the source 100 and 101 disks to the target virtual machine using the **FLASHCOPY** command. If **FLASHCOPY** is not supported or fails for some other reason, the script falls back to the **dasdfmt** and **dd** commands to copy the disks:

```
Mounting /dev/dasdi1 over /mnt/targetLinux ...
Modifying cloned image under /mnt/targetLinux ...
Removing SSH keys
Removing 9.60.18.224 from known_hosts file
Setting device 0.0.2100 offline
Done
Invoking CP command: DETACH 2100
DASD 2100 DETACHED
```

In the section above, the script mounts the newly copied root file system over a mount point, /mnt/targetLinux/. The networking information is modified and the SSH keys are regenerated. Then, the disk is detached:

```
Invoking CP command: XAUTOLOG LINUX4
Command accepted
Successfully cloned S113GOLD to LINUX4
```

In the final section, the target virtual machine is logged on using **XAUTOLOG**. Because the PROFILE EXEC detects that the ID is logged on in a disconnected mode, Linux is IPLed from minidisk 100. The new system should be in the network in about 30 - 45 seconds.

Note: If the **clone.sh** script fails, you can add the **-v** flag for some more diagnostics. Also, check the following conditions:

- The target virtual machine has been granted access to the virtual switch.
- The parameter file is copied and set correctly on the LNXMAINT 192 disk.
- The target 100 and 101 minidisks are sized identically.

Congratulations. You have successfully cloned a SUSE Linux Enterprise Server 11 SP3 system both manually and automatically. The next section describes how to create appliances from these *clones*.

17.4 System patching philosophies

When the first enterprise distribution for mainframe from SUSE appeared in 2000, a new concept was needed to provide updates and patches in a defined and predictable way. The set of rules agreed upon (that later on was adopted by many other distributions) were:

- Code errors should be fixed in the very same package version as was distributed in GA level.
- Updates are only done when really needed. The product management must decide if an update is really needed.
- The programming interfaces must stay stable within major releases.
- The source code for all architectures must be consistent. This also means that an RPM that fails QA on the mainframe architecture will also block the same patch for x86.

A big process was set up around this, including decision management of product management, the actual code fixing, quality assurance, creating documentation for every patch, and finally making defined release dates possible for fixes. The basic structure of this update process has proven solid and useful over the time and thus is still in place.

The Linux patching process on SUSE Linux Enterprise Server is different from the one on other operating systems. It does not require the system to keep a golden image, or a preproduction version. Doing that kind of processes is possible, but more a choice and task of the actual administrator.

All system files of the distribution are kept in an RPM database, which can correlates between the actual file and a package. These packages are signed with a distribution key, and thus can be checked by the administrator for validity. That way, RPM also can detect if a system file changes.

A patch for SUSE Linux Enterprise Server commonly contains one or more packages, but also might just contain a script or an image file. All patches are signed and can be checked for validity. The packages that are kept within a patch relate to one or more fixes, which are explained in the patch documentation.

To update all pending patches except those that require interaction, use the following command:

```
# zypper up -t patch --skip-interactive
```

When doing this in an automated way, some kind of monitoring should check if the system is up-to-date. Especially kernel updates still require a reboot and are not installed automatically with the preceding command.

18

Create SUSE Linux Enterprise Server 11 SP3 appliances

"*Common sense is the collection of prejudices acquired by age eighteen.*"

— Albert Einstein

This chapter describes how to clone and customize the following Linux virtual servers:

- "Creating a virtual web server" on page 318
- "Create a virtual LDAP server" on page 319
- "Create a virtual file server" on page 326
- "Create an application development appliance" on page 330

These Linux virtual servers can be thought of as *virtual appliances* after they have been cloned and *personalities* have been added to them.

18.1 Deploying and setting up services on SUSE Linux Enterprise Server

After the installation of a Linux on System z, the remaining tasks to set up some service are very similar to the ones that have to be done on any other platform for SUSE Linux Enterprise Server. Even the available packages are normally built from the identical source code.

The system management tool YaST provides a number of modules to install and configure services. Setting up servers for services like DHCP, DNS, FTP, HTTP, IUCV Terminal, Printer, and many more are available for your use. This makes it very easy to get started with, and the defaults commonly use reasonable values. Some short help is commonly also available from within the respective module.

For deployments that are more specific, or if an explanation of concepts needed, see the following website:

https://www.suse.com/documentation/sles11

As a common rule, you can find information about a certain package if you use the following sources. As an example, the package **apache** is used:

- Find the available packages with apache in name:

```
# zypper search apache
# zypper search apache
Loading repository data...
Reading installed packages...

S | Name                  | Summary                                        | Type
--+-----------------------+------------------------------------------------+-----------
  | apache2               | The Apache Web Server Version 2.2              | package
  | apache2               | The Apache Web Server Version 2.2              | srcpackage
  | apache2-doc           | Additional Package Documentation.              | package
  | apache2-example-pages | Example Pages for the Apache 2 Web Server      | package
  | apache2-mod_apparmor  | AppArmor module for apache2                    | package
  | apache2-mod_auth_kerb | Kerberos Module for Apache                     | package
  | apache2-mod_auth_kerb | Kerberos Module for Apache                     | srcpackage
  | apache2-mod_jk        | Connectors between Apache and Tomcat Servl->   | package
  | apache2-mod_jk        | Connectors between Apache and Tomcat Servl->   | srcpackage
  | apache2-mod_macro     | Define and Use Macros within the Apache Co->   | package
  | apache2-mod_macro     | Define and Use Macros within the Apache Co->   | srcpackage
  | apache2-mod_mono      | Run ASP.NET Pages on Unix with Apache and ->   | package
  | apache2-mod_mono      | Run ASP.NET Pages on Unix with Apache and ->   | srcpackage
  | apache2-mod_perl      | Embedded Perl for Apache                       | package
  | apache2-mod_perl      | Embedded Perl for Apache                       | srcpackage
  | apache2-mod_php53     | PHP5 Module for Apache 2.0                     | package
  | apache2-mod_python    | A Python Module for the Apache 2 Web Server    | package
  | apache2-mod_python    | A Python Module for the Apache 2 Web Server    | srcpackage
  | apache2-mod_security2 | ModSecurity Open Source Web Application Fi->   | package
  | apache2-mod_security2 | ModSecurity Open Source Web Application Fi->   | srcpackage
  | apache2-prefork       | Apache 2 "prefork" MPM (Multi-Processing M->   | package
  | apache2-utils         | Apache 2 utilities                             | package
  | apache2-worker        | Apache 2 worker MPM (Multi-Processing Modu->   | package
```

- The documentation directory is /usr/share/doc/packages/<package>. For example, to see the documentation for `vim`, do the following action:

```
# ls /usr/share/doc/packages/vim
ANNOUNCEMENT.vim-7.2  README_lang.txt  suse.vimrc
INSTALL               README_src.txt   vimrc_example1
README.txt            README_unix.txt  vimrc_example2
```

- Find the URL of the original project of the package. This is commonly available in the RPM header as URL:

```
# rpm -qi vim
Name        : vim                          Relocations: (not relocatable)
Version     : 7.2                               Vendor: SUSE LINUX Products GmbH,
Nuernberg, Germany
Release     : 8.15.2                        Build Date: Sat Feb  4 05:04:44 2012
Install Date: Mon Jul 15 15:43:08 2013         Build Host: s390z17
Group       : Productivity/Editors/Vi       Source RPM: vim-7.2-8.15.2.src.rpm
Size        : 2070560                          License: Other uncritical OpenSource
License; "http://vimdoc.sourceforge.net/htmldoc/uganda.html#license ...";
http://vimdoc.sourceforge.net/htmldoc/uganda.html#license
Signature   : RSA/8, Sat Feb  4 05:05:36 2012, Key ID e3a5c360307e3d54
Packager    : http://bugs.opensuse.org
URL         : http://www.vim.org/
Summary     : Vi IMproved
Description :
Vim (Vi IMproved) is an almost compatible version of the UNIX editor
```

```
vi. Almost every possible command can be performed using only ASCII
characters. Only the 'Q' command is missing (you do not need it). Many
new features have been added: multilevel undo, command line history,
file name completion, block operations, and editing of binary data.

Vi is available for the AMIGA, MS-DOS, Windows NT, and various versions
of UNIX.

For SuSE Linux, Vim is used as /usr/bin/vi.

Package vim contains the normal version of vim. To get the full runtime
environment install additionally vim-data.



Authors:
--------
    Bram Moolenaar <bram@vim.org>
Distribution: SUSE Linux Enterprise 11
```

You might want to open the main URL in a browser:

http://www.vim.org

- Find the location of the configuration files and binary files of the package:

```
# rpm -ql vim
/bin/vim-normal
```

- Check if some manual or information pages are available.

When doing deployments of software, it is good practice to spend some time considering where to put the respective parts of the software. If this is some package from SUSE, the locations are already added to the RPM and commonly no problems occur. When deploying software from other sources, it is good to have an idea, which directories have a special meaning and should be considered for a dedicated use. The File Hierarchy Standard (FHS), which is part of the Linux Standard Base (LSB) is very helpful in this respect. The specification of this is found on the web at http://www.pathname.com/fhs/pub/fhs-2.3.html. A very short overview about important directories for administrators is found in table Table 18-1.

Table 18-1 Important directories

Directory	Special functions	Comment
/tmp/	Temporary content only. This content might vanish after a reboot.	Every user can write to this directory. Access rights still apply.
/var/tmp/	Temporary content that will survive reboots.	Similar as /tmp.
/usr/local/	Private space for the administrator. SLES is not going to touch anything there during updates or installations.	This is a good place for system-wide scripts that are done for this system only. This also has its on directory tree as in subdirectories for **bin**, **etc**, **lib**, and so on.
/opt/	Space for third-party applications. For example, /opt/IBM is used often for some software distributed by IBM.	

Directory	Special functions	Comment
`/srv/`	All data that should be made available to the outside world should stay here.	All NFS, FTP, WWW, SMB, and other data for the network should go there.
`/etc/`	Configuration files.	Services the need some kind of configuration will have their configuration either in /etc or in a subdirectory of /etc.

18.2 Creating a virtual web server

The example in this section uses the `LINUX153` virtual machine to create a virtual Web server. You should have a vanilla virtual server cloned to the virtual machine `LINUX153` as described in Chapter 17, “Clone SUSE Linux Enterprise Server 11 SP3” on page 305.

18.2.1 Installing Apache RPMs

This section describes how to install the Apache Web Server RPMs. To do so, perform the following steps:

1. **SSH** as root to the Linux system running on a Linux clone. In this example, ***LINUX4*** is used.
2. Install the following Apache RPMs using the **`zypper install`** command:

```
# zypper install apache2-prefork apache2 apache2-doc apache2-example-pages
zypper install apache2-prefork apache2 apache2-doc apache2-example-pages
Loading repository data...
Reading installed packages...
Resolving package dependencies...

The following NEW packages are going to be installed:
  apache2 apache2-doc apache2-example-pages apache2-prefork apache2-utils
  libapr-util1 libapr1

The following recommended package was automatically selected:
  apache2-prefork

7 new packages to install.
Overall download size: 3.1 MiB. After the operation, additional 14.1 MiB will
be used.
Continue? [y/n/?] (y): y
...
```

3. Confirm the RPMs have been added using the **`rpm -qa`** command:

```
# rpm -qa | grep apache
apache2-doc-2.2.12-1.38.2
apache2-utils-2.2.12-1.38.2
apache2-example-pages-2.2.12-1.38.2
apache2-2.2.12-1.38.2
apache2-prefork-2.2.12-1.38.2
```

You now have the Apache RPMs installed.

18.2.2 Testing Apache

Start the Apache Web Server to verify that it is installed successfully:

1. Start the Apache server using the **service** command and set it to start at boot time using the **chkconfig** command:

```
# service apache2 start
Starting httpd2 (prefork)                                done
# chkconfig apache2 on
```

2. To verify that Apache is installed correctly, after it has been started, point a web browser to the server and see the Apache test page. In your web browser, put on the host name or IP address of your Web server as the URL. In this example, the virtual server running on LINUX4 has a host name of virtcook.ibm.com:

```
http://virtcook4.itso.ibm.com
```

3. You should see a test page with two words: **It works!**

If you get an error in starting Apache, look in the log file /var/log/apache2/error-log for clues. If Apache started successfully but you cannot reach the test page from a browser, try accessing it using the IP address rather than the DNS name.

18.2.3 Populating your website

You can begin to put your web pages in the directory /srv/www/htdocs/, which is the default Web root. For security and customization purposes, you might want to change the default Web root to point to another directory. The easiest way to do this is to copy /etc/apache2/default-server.conf to your own configuration file, that is, /etc/apache2/my-server.conf. Make the changes in /etc/apache2/my-server.conf, and then edit /etc/apache2/httpd.conf to use my-server.conf.

18.2.4 Apache resources

The following websites contain additional information about Apache:

- http://www.samspublishing.com/articles/article.asp?p=30115&seqNum=4
- http://www.sitepoint.com/article/securing-apache-2-server-ssl
- http://www.securityfocus.com/infocus/1786

18.3 Create a virtual LDAP server

The Lightweight Directory Access Protocol (LDAP) is commonly implemented using the OpenLDAP package, which comes standard with most Linux distributions. Among other directory functions, OpenLDAP allows for centralized login authentication and user and group ID resolution.

In this section, you will clone Linux and configure a new virtual LDAP server. Then, you will go back to the virtual Web server that you just created and point it to the new LDAP server.

Then, you might want to configure the golden image so that it is pointing to this virtual server. If you do so, all Linux images that are cloned will be able to use this virtual LDAP server.

The following steps are covered in this section:

1. "Clone a Linux" on page 320

2. “Configure the LDAP server” on page 320
3. “Add an LDAP user” on page 322
4. “Set another virtual server to use the LDAP server” on page 323

18.3.1 Clone a Linux

From a root session on the Linux administration system, clone from the golden image (S113GOLD virtual machine) to LINUX5 using the **clone.sh** script:

```
# clone.sh from s113gold to linux5
Checking that S113GOLD exists and is not logged on ...
Invoking CP command: QUERY S113GOLD
HCPCQU045E S113GOLD not logged on
Error: non-zero CP response for command 'QUERY S113GOLD': #45
Checking that LINUX5 exists and is not logged on ...
Invoking CP command: QUERY LINUX5
HCPCQU045E LINUX5 not logged on
Error: non-zero CP response for command 'QUERY LINUX5': #45

WARNING!!: Minidisks 100 and 101 will be copied to LINUX5
Network data is retrieved from LINUX5 PARM-S11 on 191 disk
during the first boot of LINUX5
Are you sure you want to overwrite these disks (y/n): y
...
```

The **clone.sh** script should create a new virtual server.

18.3.2 Configure the LDAP server

To configure the OpenLDAP server, the **yast** tool is recommended:

1. Start an SSH session as root to the new server.
2. Invoke the **yast** command. The YaST Control Center should be displayed:

```
# yast
```

3. Use the down arrow key to move to *Network Services* on the left side. Use the Tab or right arrow key to move to the right side and select *LDAP Server.* Press **Enter**.

```
+-----------------------------------------------------------------------+
| YaST2 Control Center                                                  |
+-----------------------------------------------------------------------+
|Software               - -DHCP Server                                  |
|Hardware               - -DNS Server                                   |
|System                 - -FTP Server                                   |
|Network Devices        - -HTTP Server                                  |
|Network Services       - -Hostnames                                    |
|Security and Users     - -IUCV Terminal Server                         |
|Virtualization         - -IUCV Terminals                               |
|Support                - -Kerberos Client                              |
|Miscellaneous          - -Kerberos Server                              |
|                       - -LDAP Browser                                 |
|                       - -LDAP Client                                  |
|                       - -LDAP Server                                  |
|                       - -Mail Server                                  |
...
```

4. You should see a pop-up window with the following message. Press **Enter** to accept the default of **Install** and some RPMs will be installed:

```
These packages need to be installed: openldap2
```

5. The *LDAP Server Configuration* panel is displayed. Accept the default of **Yes** by using the tab key to **Next** and pressing **Enter**:

```
                     General Settings
+Start LDAP Server-------------------------------+
|(x) Yes                                         |
|( ) No                                          |
|[ ] Register at an SLP Daemon                   |
+------------------------------------------------+

+Firewall Settings-------------------------------+
|[ ] Open Port in Firewall  [Firewall Details...]|
|Firewall is disabled                            |
+------------------------------------------------+
```

6. A second *LDAP Server Configuration* panel will appear. Accept the default of **Standalone server** by using the tab key to **Next** and pressing **Enter**.

```
                           Server type
+----------------------------------------------------------------------------+
¦                                                                            ¦
¦(x) Stand-alone server                                                      ¦
¦                                                                            ¦
¦( ) Master server in a replication setup                                    ¦
¦                                                                            ¦
¦( ) Replica (slave) server.                                                 ¦
¦    All data, including configuration, is replicated from a remote server.¦
+----------------------------------------------------------------------------+
```

7. In the *TLS settings* panel, accept all defaults of using TLS and press **Next**. This ensures that LDAP communications are encrypted.
8. The *Basic Database Settings* panel appears. The *Base DN* (in this example `dc=itso,dc=ibm,dc=com`) should be correct for your DNS domain. Set the LDAP administrator password (twice) and select **Next**:

```
New Database
                          Basic Database Settings
         Database Type
         hdb#########â
         Base DN
         dc=itso,dc=ibm,dc=com####################################

         Administrator DN
         cn=Administrator###################### [x] Append Base DN

         LDAP Administrator Password
         ******###################################################
         Validate Password
         ******###################################################

         Database Directory
         /var/lib/ldap################################ [Browse...]
         [x] Use this database as the default for OpenLDAP clients
```

9. The *LDAP Server Configuration Summary* panel should appear now, this time with one database identified by the Base DN (`dc=itso,dc=ibm,dc=com` in this example). Select **Finish** by pressing **Enter**.
10. The panel *Saving LDAP Server Configuration* should appear. The database will be created and the LDAP server configured.
11. You should be returned to the *YaST Control Center*. Move the cursor to **Quit** and press **Enter**.
12. Verify that the LDAP server is running with the **service** command and that it is set to start in run levels 3 and 5 with the **chkconfig** command:

```
# service ldap status
Checking for service ldap:                                           running
# chkconfig --list ldap
ldap                      0:off  1:off  2:off  3:on   4:off  5:on   6:off
```

You have now cloned a new virtual server and configured it to run OpenLDAP.

18.3.3 Add an LDAP user

When the golden image was installed, it was recommended that a non-root user be added. In this example, it was named `mikemac`:

1. Verify that this user exists with the **id** command and see that there is an entry in the `/etc/passwd` file with the **grep** command:

```
# id mikemac
uid=1000(mikemac) gid=100(users) groups=16(dialout),33(video),100(users)
# grep mikemac /etc/passwd
mikemac:x:1000:100::/home/mikemac:/bin/bash
```

2. Delete this local user using the **userdel** command so it can be added to LDAP later:

```
# userdel mikemac
no crontab for mikemac
# id mikemac
id: mikemac: No such user
```

3. An LDIF (LDAP Interchange Format) file is created to add an organizational unit named **People** and a user named ***mikemac***. Create a similar file for your system's values:

```
# cd /var/lib/ldap
# vi initial.ldif // create the input file ...
dn: ou=People,dc=itso,dc=ibm,dc=com
ou: People
objectClass: top
objectClass: organizationalUnit

dn: uid=mikemac,ou=People,dc=itso,dc=ibm,dc=com
uid: mikemac
cn: mikemac
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
loginShell: /bin/bash
uidNumber: 501
gidNumber: 100
homeDirectory: /home/mikemac
```

4. Add the contents of the LDIF file to the LDAP server with the **ldapadd** command (the line wraps, but it is one command):

```
# ldapadd -x -h localhost -D "cn=Administrator,dc=itso,dc=ibm,dc=com" -f initial.ldif -W
Enter LDAP Password:
adding new entry "ou=People,dc=itso,dc=ibm,dc=com"

adding new entry "uid=mikemac,ou=People,dc=itso,dc=ibm,dc=com"
```

5. Search for the new user just added with the **ldapsearch** command:

```
# ldapsearch -x uid=mikemac
# extended LDIF
#
# LDAPv3
# base <dc=itso,dc=ibm,dc=com> (default) with scope subtree
# filter: uid=mikemac
# requesting: ALL
#

# mikemac, People, itso.ibm.com
dn: uid=mikemac,ou=People,dc=itso,dc=ibm,dc=com
uid: mikemac
cn: mikemac
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
loginShell: /bin/bash
uidNumber: 501
gidNumber: 100
homeDirectory: /home/mikemac

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

   This shows that the user exists in the LDAP database.

6. Set the password with the **ldappasswd** command. You will need to provide a new password for the new user and you will also need to provide the LDAP administrator password:

```
# ldappasswd -x -D "cn=Administrator,dc=itso,dc=ibm,dc=com" -W -S
"uid=mikemac,ou=People,dc=itso,dc=ibm,dc=com"
New password:
Re-enter new password:
Enter LDAP Password:
```

You have now deleted a local user, added a new LDAP user using an LDIF file, and have set the new LDAP user's password.

18.3.4 Set another virtual server to use the LDAP server

Now that you have a virtual LDAP server, you might want to point another virtual server to it so that you will have a centralized user database. If you have been following along in this book, you should have created a Web server running on the ***LINUX4*** virtual machine.

To point it to an LDAP server is fairly easy. In this section you perform the following steps:

1. "Test that the LDAP client is not working"
2. "Use YaST to modify the LDAP authentication client"
3. "Test the LDAP client" on page 325

Test that the LDAP client is not working

Before you start, try a couple of commands to show that LDAP is *not* working. To do so, perform the following steps:

1. Start an SSH session as root to the virtual web server. In this example, it is ***LINUX4***.
2. Search for the LDAP user that you added earlier to the virtual LDAP server. In this example, it is `mikemac`:

```
# ldapsearch -x uid=mikemac
ldap_sasl_bind(SIMPLE): Can't contact LDAP server (-1)
```

   The **ldapsearch** command cannot resolve the LDAP user because it cannot contact the LDAP server.

3. Delete the non-root user (`mikemac` in this example) from the local file system with the **userdel** command:

```
# userdel mikemac
no crontab for mikemac
```

Use YaST to modify the LDAP authentication client

The **yast** system administration interface can be used to configure the LDAP authentication client:

1. Invoke the **yast** command. The `YaST Control Center` should appear:

```
# yast
```

2. Select **Network Services** on the left side and **LDAP Client** on the right. Press **Enter**:

```
+----------------------------------------------------------------------+
|                          YaST2 Control Center                        |
+----------------------------------------------------------------------+
|Software             - -DHCP Server                                 |
|Hardware             - -DNS Server                                  |
|System               - -FTP Server                                  |
|Network Devices      - -HTTP Server                                 |
|Network Services     - -Hostnames                                   |
|Security and Users   - -IUCV Terminal Server                        |
|Virtualization       - -IUCV Terminals                              |
|Support              - -Kerberos Client                             |
|Miscellaneous        - -Kerberos Server                             |
|                     - -LDAP Browser                                |
|                     - -LDAP Client                                 |
|                     - -LDAP Server                                 |
|                     - -Mail Server
  ...
```

3. On the *LDAP Client Configuration panel*, perform the following steps:
   a. Use the Tab key to move to **Use LDAP** and press the **space bar** to select that choice.
   b. Move to the *Addresses of LDAP Servers* field and enter the IP address (or DNS name) of your LDAP server. You can either enter LDAP base DN manually or press **Fetch DN** and then **OK** in result window. This way, you can ensure that the LDAP server is accessible.

c. Clear **LDAP TLS/SSL**:

```
|User Authentication--------------------------------------------------------
 |( ) Do Not Use LDAP                                                      |
 | (x) Use LDAP                                                             |
 | ( ) Use LDAP but Disable Logins                                          |
 | [ ] Use System Security Services Daemon (SSSD)                           |
  ---------------------------------------------------------------------------
  |LDAP Client-------------------------------------------------------------|
  | Addresses of LDAP Servers                                              |
  | virtcook5.itso.ibm.com-------------------------------------------[Find] |
  | LDAP Base DN                                                           |
  | dc=itso,dc=ibm,dc=com----------------------------------------[Fetch DN] |
  |------------------------------------------------------------------------|
  |Secure Connection-------------------------------------------------------|
  | [ ] LDAP TLS/SSL                             [Download CA Certificate] |
  ---------------------------------------------------------------------------

  [ ] Start Automounter
  [ ] Create Home Directory on Login
                              [Advanced Configuration...]
```

4. Use the Tab key to move to **OK** and press **Enter**. You should get the following prompt. Press **Enter** to continue:

```
These packages need to be installed:
 pam_ldap
 nss_ldap
 pam_ldap-32bit
 nss_ldap-32bit
```

5. Accept **OK** if you get a warning window. Your changes will be saved.
6. At the main *YaST2 Control Center*, press **Quit** on the main window to quit YaST.

Your Web server virtual Linux should now also be using OpenLDAP for user and group ID resolution and authentication.

Test the LDAP client

To test the LDAP client, perform the following steps:

1. Try the **id** command against the new LDAP user:

```
# id mikemac
uid=501(mikemac) gid=100(users) groups=100(users)
```

   Note that the UID is 501 in this example (from the LDIF file), not the value 1000 (from the Linux installation).

2. Try the **ldapsearch** command again:

```
# ldapsearch -x uid=mikemac
# extended LDIF
#
# LDAPv3
# base <dc=itso,dc=ibm,dc=com> (default) with scope subtree
# filter: uid=mikemac
# requesting: ALL
#

# mikemac, People, endicott.ibm.com
dn: uid=mikemac,ou=People,dc=itso,dc=ibm,dc=com
uid: mikemac
cn: mikemac
```

```
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
loginShell: /bin/bash
uidNumber: 501
gidNumber: 100
homeDirectory: /home/mikemac

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

3. Start an SSH session to the virtual web server using the LDAP user. You should be able to successfully start a session.

You might also want to set the golden image to authenticate with the LDAP server. In this fashion, all virtual servers cloned after that will be able to use a centralized authentication server.

18.4 Create a virtual file server

Samba allows MS Windows clients to map Linux file systems as shared drives. Samba can also act as a middle-man between MS Windows clients and a Linux print server. The recommended Linux print server is the Common UNIX Printing System (CUPS). This section does *not* describe the configuration of CUPS, but it does describe how the necessary RPMs are installed.

The steps in this section are as follow:

1. "Install Samba and CUPS"
2. "Configure Samba configuration file" on page 327
3. "Add a Samba user" on page 328
4. "Set Samba to start at boot time" on page 328
5. "Test the configuration" on page 328

18.4.1 Install Samba and CUPS

To install Samba and CUPS, perform the following steps:

1. Start an SSH session to a virtual server as root. In this example, the virtual machine `LINUX5` is used again.
2. Install necessary RPMs with the following **`zypper install`** command:

```
# zypper install samba yast2-samba-server samba-doc samba-winbind cups cups-drivers
Loading repository data...
Reading installed packages...
'yast2-samba-server' is already installed.
No update candidate for 'yast2-samba-server-2.17.15-0.7.79.noarch'. The highest
available version is already installed.
Resolving package dependencies...

The following NEW packages are going to be installed:
  a2ps cups cups-drivers fam foomatic-filters ghostscript-fonts-other
```

```
  ghostscript-fonts-std ghostscript-library ghostscript-omni libavahi-client3
  libavahi-common3 libdns_sd libgimpprint libgmodule-2_0-0 libiniparser0
  libiniparser0-32bit libldb1 libpoppler5 libpython2_6-1_0-32bit libtalloc2-32bit
  libtdb1 libtdb1-32bit libtevent0 libwbclient0 libwbclient0-32bit poppler-data
  poppler-tools samba samba-32bit samba-client samba-client-32bit samba-doc
  samba-krb-printing samba-winbind samba-winbind-32bit wdiff

The following recommended packages were automatically selected:
  ghostscript-fonts-other ghostscript-omni poppler-data samba-krb-printing

36 new packages to install.
Overall download size: 69.4 MiB. After the operation, additional 300.5 MiB will be
used.
Continue? [y/n/?] (y): y
...
```

3. Confirm that the RPMs were added:

```
# rpm -qa | egrep "samba|cups"
cups-libs-32bit-1.3.9-8.46.46.1
samba-client-32bit-3.6.3-0.39.1
samba-winbind-3.6.3-0.39.1
cups-1.3.9-8.46.46.1
cups-client-1.3.9-8.46.46.1
yast2-samba-client-2.17.27-0.7.29
yast2-samba-server-2.17.15-0.7.79
samba-winbind-32bit-3.6.3-0.39.1
samba-32bit-3.6.3-0.39.1
samba-krb-printing-3.6.3-0.39.1
cups-drivers-1.3.9-2.31
cups-libs-1.3.9-8.46.46.1
samba-doc-3.6.3-0.39.1
samba-client-3.6.3-0.39.1
samba-3.6.3-0.39.1
```

The Samba and CUPS RPMs are now installed.

18.4.2 Configure Samba configuration file

The one configuration file for Samba is `/etc/samba/smb.conf`. It is easy to add an SMB share that will be made available by the Samba server. A good test directory is `/usr/share/doc/` because it has excellent 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 = SLES 11 SP3 on System z documentation
        path = /usr/share/doc/
```

This will cause an SMB share named **sharedoc** consisting of the contents of `/usr/share/doc` to be created when Samba is started.

18.4.3 Add a Samba user

The default method that Samba uses to determines users' credentials is to look in the /etc/samba/smbpasswd file. That user must first exist in the Linux file system (/etc/passwd, /etc/shadow, and so on).

The following example shows adding the user mikemac to the smbpasswd file. Create a new Samba user with the **smbpasswd -a** command:.

```
# smbpasswd -a mikemac
New SMB password:
Retype new SMB password:
Added user mikemac.
```

This method of maintaining Samba users, groups, and passwords is good for a few 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 18.3.4, "Set another virtual server to use the LDAP server" on page 323 because the Samba schema must first be added to LDAP.

18.4.4 Set Samba to start at boot time

Samba consists of two daemons **nmbd** and **smbd**. To set Samba to start at Linux boot time, perform the following steps:

1. Start Samba in the current session with the **service** command for both the **nmb** and **smb** daemons:

```
# service nmb start
Starting Samba NMB daemon                                              done
# service smb start
Starting Samba SMB daemon                                              done
```

2. Use the **chkconfig** command in order to start these daemons at boot time:

```
# chkconfig nmb on
# chkconfig smb on
```

Samba should now be running and configured to start at boot time.

18.4.5 Test the configuration

To test the Samba configuration, perform the following steps:

1. Verify that the Samba daemons are running with the **service** command:

```
# service nmb status
Checking for Samba NMB daemon                                          running
# service smb status
Checking for Samba SMB daemon                                          running
```

2. Test getting a Samba share from a Windows desktop:

   a. Go to any Windows Explorer window (such as *My Computer*) and select **Tools** -> **Map Network Drive**.

   b. Use the Universal Naming Convention (UNC) to specify the Samba server and share name as shown in the upper left corner of Figure 18-1 on page 329. In this example, the UNC is ***9.12.7.5*\sharedoc**.

c. You might need to click **Connect using different user name**, if the sample virtual machine and password are different on your desktop computer from the values you set on the Samba server.

d. Click **Finish**. If all the steps were correct, you should see the files in a new Explorer window, as shown in the lower right corner of the figure.

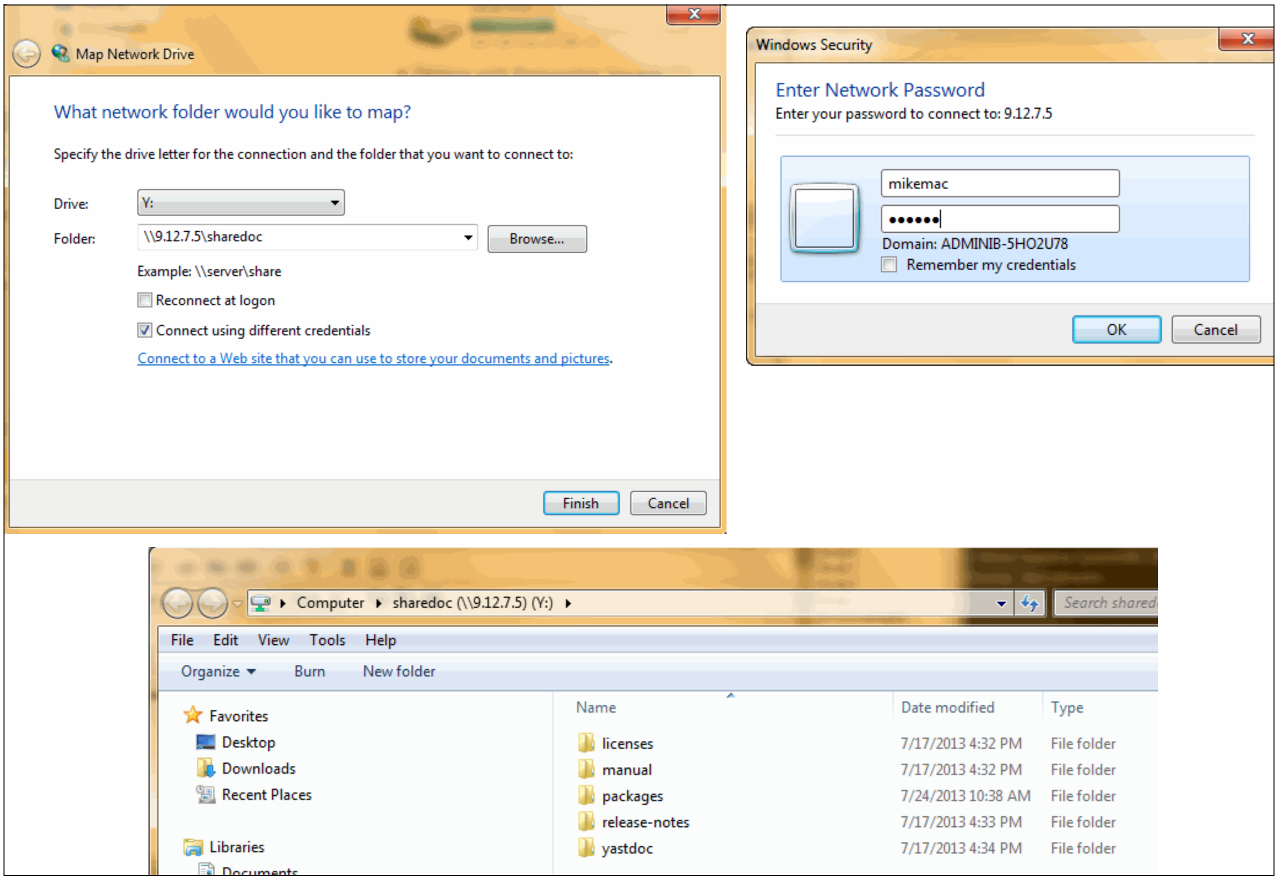


Figure 18-1 Mapping a network drive to a Samba share

You should now have Samba configured and running with one new share available.

1. Test getting a Samba share from a Windows DOS session:

a. Use the **NET USE** command to access a Samba share:

```
c:\> net use * \\9.12.7.5\sharedoc
Drive Y: is now connected to \\9.12.7.5\sharedoc.

The command completed successfully.
```

b. Query the Samba server with the **NET VIEW** command:

```
c:\> net view \\9.12.7.5
Shared resources at \\9.12.7.5

Samba 3.6.3-0.39.1-3012-SUSE-CODE11-s390x

Share name  Type  Used as  Comment

-------------------------------------------------------------------------------
groups      Disk           All groups
mikemac     Disk           Home Directories
```

```
profiles    Disk           Network Profiles Service
sharedoc    Disk  Y:       SLES 11 SP3 on System z documentation
users       Disk           All users
The command completed successfully.
```

This section has shown how to install Samba and CUPS, and how to use Samba. Using CUPS is outside the scope of this section.

18.5 Create an application development appliance

Most Linux distributions come with a robust set of application development tools, making Linux one of the most versatile development systems. These basic tools are ideal for projects of any size.

There are three main areas of development in Linux:

- Linux kernel development (C) for the Linux operating system itself, such as subsystems, device drivers, memory management.
- Application development (C/C++ and Java) for software to be used on Linux.
- Web development for applications to be run on the web, such as stock trade applications or email applications.

The development languages used in implementation range from scripting languages such as Python or Tcl, to compiled languages such as C/C++ and Java. There is software available on Linux to help form a development system for developers to create integrated applications. MySQL and Apache are among them. A popular open source web platform is LAMP, which stands for the open source software and programming languages used to make up the platform: Linux, Apache, MySQL, Python, or PHP. Other times, it is just as useful to know about Linux development tools when you want to build an application from source code downloaded from `www.sourceforge.net`.

To create an application development server, perform the following steps:

1. **Start an SSH session as root** to the new virtual server.
2. You can install *all* the application development tools described in this section with the following command:

```
# zypper install python perl tcl php gcc gdb make java-1_7_0-ibm
...
15 new packages to install.
Overall download size: 78.6 MiB. After the operation, additional 194.2 MiB will be
used.
...
```

If you want to install only certain tools, each specific RPM or group of RPMs is described in the sections that follow.

18.5.1 Scripting languages

Scripts are good for quickly automating a process or writing your own commands. They are also used for being the backbone of robust applications. There are numerous scripting languages used in Linux application development; here are overviews of the most popular and general ones, obtained from their package descriptions:

- **Python**: Python is an interpreted, object-oriented programming language, and is often compared to Tcl, Perl, Scheme, or Java. You can find an overview of Python in the documentation and tutorials included in the python-doc (HTML) or python-doc-pdf (PDF) packages. To install the python interpreter, execute the following command:

```
# zypper install python
...
```

- **Perl**: Practical Extraction and Report Language. Perl is optimized for scanning arbitrary text files, extracting information from those text files, and printing reports based on that information. It is also good for many system management tasks. Perl is intended to be practical (easy to use, efficient, and complete) rather than beautiful (tiny, elegant, and minimal). To install Perl, execute the command:

```
# zypper install perl
...
```

- **tcl**: the "Tool Command Language", is a very simple programming language. Tcl provides basic language features such as variables, procedures, and control. Another very popular extension is *Expect*, which can be used to automate console-based interactive applications. To install tcl, execute the following command:

```
# zypper install tcl
...
```

- **PHP**: PHP (recursive acronym for "PHP: Hypertext Preprocessor") is a widely used Open Source general-purpose scripting language that is especially suited for web development and can be embedded into HTML. PHP development is focused on server-side scripting, but you can do much more with it. To install PHP, execute the following command:

```
# zypper install php
...
```

18.5.2 C/C++ development tools

Most Linux distributions come with the C/C++ compiler, gcc. This is also known as the *GNU Compiler Collection* (GCC) because it can compile other languages, such as Fortran, but it is most frequently used to compile C and C++ code. In the minimal SLES10 installation, none of the development packages are installed. In order to use gcc, you must install it using yast:

```
# zypper install gcc
...
# rpm -qa | grep gcc
libgcc46-4.6.9-0.11.38
libgcc43-32bit-4.6.9-0.11.38
libgcc_s1-4.7.2_20130108-0.15.45
gcc43-4.3.4_20091019-0.37.30
gcc-4.3-62.198
libgcc43-4.6.9-0.11.38
libgcc_s1-32bit-4.7.2_20130108-0.15.45
libgcc46-32bit-4.6.9-0.11.38
```

The GNU debugger, or **gdb**, is a popular and robust debugger for C/C++ programs. You can step through your program (that has been successfully compiled) to see where it is failing. Install it using yast:

```
# zypper install gdb
...
```

Keep in mind that you can also set breakpoints at functions in the code. Refer to the manual page of **gdb** for more information: **man gdb**.

To make a large program more manageable, developers usually create a makefile that specifies instructions on how to compile a program. Then, use the GNU **make** tool to use the makefile to make a working program. To install `make`, issue the following command:

```
# zypper install make
...
```

18.5.3 Java development tools

SUSE Linux Enterprise Server 11 comes with IBM Java *Standard Development Kit* (*SDK*), which is needed if you want to develop Java applications. You need a Java Runtime Environment (JRE) if you want to only run Java applications. The following command installs both:

```
# zypper install java-1_7_0-ibm
...
```

A good Java debugger is **jdb**. It comes with IBMJava2-SDK and can be run similar to **gdb**. A good tutorial is on the web at the following site:

http://publib.boulder.ibm.com/infocenter/realtime/v1r0/index.jsp?topic=%2Fcom.ibm.rt.doc.10%2Fuser%2Fjdb.html

You can use the GNU **make** to build from Java makefiles or the more recent and popular *Ant*. Ant uses XML technology. Here is a great guide to get you started with either tool:

http://www.onlamp.com/pub/a/onlamp/2004/11/18/gnumake_3e.html

18.6 Setting up subversion

If you need to set up a source code control system, subversion, or *svn*, is a good choice. To set up **svn**, perform the following steps:

1. Install the subversion RPM on SUSE Linux Enterprise Server 11 SP3, perform the following step:

   ```
   # zypper install subversion-server
   ...
   ```

2. Set up an `svn` group and user with both the *gid* and *uid* being `500`:

   ```
   # groupadd -g 500 svn
   # useradd -m -c "SVN" -d /srv/svn/repos -g 500 -u 500 -s /sbin/nologin svn
   # mkdir -p /srv/svn/repos
   # chown svn:svn /srv/svn/repos
   ```

3. On SLES 11 SP3, keep the default location for the repositories in the `/etc/sysconfig/svnserve` configuration file to point to the `/srv/svn/repos/` directory and remove the **-R** (read-only) flag. Then, turn the **svnserve** service on:

   ```
   # cd /etc/sysconfig
   # vi svnserve
   ## Path:        Network/Subversion/svnserve
   ## Description: Basic configuration for svnserve

   ## Type:        string
   ## Default      "-d r /srv/svn/repos"
   #
   # Default options for the svnserve process.
   # The -R option enforces read-only access, i.e. write operations to the
   # repository (such as commits) will not be allowed.
   ```

```
# Authentication should be configured before allowing write access.
# See
http://svnbook.red-bean.com/en/1.5/svn.serverconfig.svnserve.html#svn.serverconfig.svnse
rve.auth
#
SVNSERVE_OPTIONS="-d -r /srv/svn/repos"
...
# rcsvnserve start
Starting svnserve                                                    done
# chkconfig svnserve on
```

4. Check that the server is now listening:

```
# netstat -l | grep svn
tcp        0      0 *:svn                   *:*                     LISTEN
```

5. Change into the repos directory:

```
# cd /srv/svn/repos
```

6. Create a new repository (**foo** in this example) with the **svnadmin create** command:

```
# svnadmin create foo
# cd foo
```

7. Make the `svn` user and group owner of all new directories:

```
# chown -R svn:svn /srv/svn/repos
```

8. Set up the configuration file for the new project for local access. In the `conf/svnserve.conf` file, uncomment and set **anon-access** to none, and uncomment the **auth-access** and **passwd-db** lines:

```
# cd conf
# vi svnserve.conf
### This file controls the configuration of the svnserve daemon, if you
### use it to allow access to this repository.  (If you only allow
### access through http: and/or file: URLs, then this file is
### irrelevant.)

### Visit http://subversion.tigris.org/ for more information.

[general]
### These options control access to the repository for unauthenticated
### and authenticated users.  Valid values are "write", "read",
### and "none".  The sample settings below are the defaults.
anon-access = none
auth-access = write
### The password-db option controls the location of the password
### database file.  Unless you specify a path starting with a /,
### the file's location is relative to the directory containing
### this configuration file.
### If SASL is enabled (see below), this file will NOT be used.
### Uncomment the line below to use the default password file.
password-db = passwd
...
```

9. Add a user and password. In this example it is **mikemac** and **passwd**:

```
# vi passwd
### This file is an example password file for svnserve.
### Its format is similar to that of svnserve.conf. As shown in the
### example below it contains one section labelled [users].
### The name and password for each user follow, one account per line.

[users]
```

```
mikemac = passwd
...
```

10.Set key-based or passwordless authentication for the local host. First, note that a password is needed to SSH to the local host:

```
# ssh localhost hostname
The authenticity of host 'localhost (::1)' can't be established.
RSA key fingerprint is 2b:6c:1a:0a:9f:d5:8f:22:0a:63:15:bd:60:c7:d6:99.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'localhost' (RSA) to the list of known hosts.
Password:
linux5
```

11.Copy the SSH DSA public key to the `authorized_keys` file and verify that key-based authentication is working:

```
# cd ~/.ssh
# ls
id_dsa  id_dsa.pub  known_hosts
# cat id_dsa.pub > authorized_keys
# ssh localhost hostname
linux5
```

The subversion server should now be configured with one empty project - `foo` in this example.

Populate the project

The project is now ready to be populated with files. The suggested structure of a subversion repository is like this:

```
foo
|-- branches
|-- tags
`-- trunk
    |-- dir1
    `-- dir2
```

To create this structure, perform the following steps:

1. Create directories for development. In this example - in this case under `/root/foo`. The example package has five root directories where the code resides when installed on a target Linux system:

```
# cd /root
# mkdir -p foo/trunk/etc
# mkdir -p foo/trunk/usr/local/sbin
# mkdir -p foo/trunk/usr/local/share/foo
# mkdir -p foo/trunk/srv/www/cgi-bin
# mkdir -p foo/trunk/usr/src/packages/SPECS
# cd foo/trunk/etc
# scp <hostWithCode>:/etc/foo.conf .
...     # and populate the other directories
# cd
# ls -R foo
foo:
branches  tags  trunk

foo/branches:

foo/tags:

foo/trunk:
```

```
www
...
```

The initial source code is now populated in the project.

2. Import the source code into subversion under the `/root/foo/` directory:

```
# svn import /root/foo file:///srv/svn/repos/foo/ -m "Initial import of files"
Adding         /root/foo/trunk
Adding         /root/foo/trunk/usr
Adding         /root/foo/trunk/usr/src
Adding         /root/foo/trunk/usr/src/packages

...
Committed revision 1.
```

3. The files have been copied to this server, then pulled in to subversion. However, subversion does not know about the `/root/foo/` directory that is evidenced by the fact that there are no `.svn/` subdirectories. Remove the unversioned files from the `/root/foo/` directory and check the files back out:

```
# rm -fr /root/foo/*
# svn co --username mikemac svn://localhost/foo /root/foo
Authentication realm: <svn://localhost:3690> a2ec81fa-d837-11e2-b5ce-55828dfb997a
Password for 'mikemac':

-----------------------------------------------------------------------
ATTENTION!  Your password for authentication realm:

   <svn://localhost:3690> a2ec81fa-d837-11e2-b5ce-55828dfb997a

can only be stored to disk unencrypted!  You are advised to configure
your system so that Subversion can store passwords encrypted, if
possible.  See the documentation for details.

You can avoid future appearances of this warning by setting the value
of the 'store-plaintext-passwords' option to either 'yes' or 'no' in
'/root/.subversion/servers'.
-----------------------------------------------------------------------
Store password unencrypted (yes/no)? yes
A    /root/foo/trunk
A    /root/foo/trunk/usr
A    /root/foo/trunk/usr/src
...
```

Note that `.svn/` subdirectories have now been created in each directory:

```
# find /root/foo -name .svn
/root/foo/trunk/usr/src/packages/SPECS/.svn
/root/foo/trunk/usr/src/packages/.svn
/root/foo/trunk/usr/src/.svn
/root/foo/trunk/usr/local/sbin/.svn
...
```

This shows that a subversion repository has been set up and populated.

18.7 Create an RPM

When it comes time to distribute your code, you may want to create and RPM, as this is a common Linux package format. An RPM for a package can be created through the combination of checking code out of SVN, the **rpm-build** package, and a *spec file* for the package.

With SLES, all packages are normally built in a special environment, only created for that one package build. This procedure ensures, that packages are always consistently build against a defined repository and eliminates problems with developers having different build machines.

If the package is to be built on the same architecture as the local developer machine, the package can also be built locally with the help of the script build.

The package build will parse the spec file for additional requirements and set up a build environment accordingly. It will then start the build process and log progress as well as the different location of generated RPMs on the console:

1. Install the `rpm` **build** on SUSE Linux Enterprise Server 11 SP3:

   ```
   # zypper install build
   ...
   ```

2. Write, get, or check out some source rpm package. This package will commonly be put into a subdirectory for each package and at least contains a **.spec** file. Most packages will also have some sort of sources that are processed by the rules in the spec file. To actually build the package **tree** from the distribution, proceed as follows:

   a. Copy tree-1.5.1-2.8.src.rpm from the second DVD to the build machine. This file is found below /suse/src.

   b. Create a subdirectory tree and change into there:

   ```
   # mkdir -p foo/trunk/tree
   # cd foo/trunk/tree
   ```

   c. Retrieve all the files that are available in the source rpm:

   ```
   # unrpm /root/tree-1.5.1-2.8.src.rpm
   /root/tree-1.5.1-2.8.src.rpm:   63 blocks
   ```

   d. Run the actual build process:

   ```
   # build tree.spec
   logging output to /var/tmp/build-root/.build.log...
   Memory limit set to 671984KB
   Using BUILD_ROOT=/var/tmp/build-root
   Using BUILD_ARCH=s390x


   linuxadmin started "build tree.spec" at Tue Jun 18 18:21:42 UTC 2013.


   processing specfile /root/foo/trunk/tree/tree.spec ...
   init_buildsystem --cachedir /var/cache/build --repository zypp://
   /root/foo/trunk/tree/tree.spec ...
   initializing /var/tmp/build-root/.srcfiles.cache ...
   /usr/lib/build/createrepomddeps
   zypp://SUSE-Linux-Enterprise-Server-11-SP3_11.3.3-1.138_1
   /usr/lib/build/createrepomddeps
   zypp://SUSE-Linux-Enterprise-Software-Development-Kit-11-SP3_11.3.3-1.69
   expanding package dependencies...
   ```

```
Warning: distribution not specified, assuming 'sles11sp2-s390x' (see
/usr/lib/build/configs).
preinstalling aaa_base...
preinstalling acl...
preinstalling attr...
preinstalling bash...
...
RPMLINT report:
===============
tree.s390x: W: unstripped-binary-or-object /usr/bin/tree
2 packages and 0 specfiles checked; 0 errors, 1 warnings.

linuxadmin finished "build tree.spec" at Tue Jun 18 18:22:02 UTC 2013.
```

e. The resulting rpm for the previous build is found below the BUILDROOT, which by default is set to `/var/tmp/build-root`. Together with the file name `/usr/src/packages/RPMS/s390x/tree-1.5.1-2.8.s390x.rpm`, the resulting rpm is at:

```
/var/tmp/build-root/usr/src/packages/RPMS/s390x/tree-1.5.1-2.8.s390x.rpm
```

f. All of the output that probably scrolled too fast to watch is available at */var/tmp/build-root/.build.log*.

g. It is possible to look inside the buildroot with the following commands:

```
# chroot --userspec=abuild:abuild
> cd /usr/src/packages/BUILD
```

18.7.1 Additional resources

The following websites are resources for additional application development information:

Scripting languages

- http://www.perl.com
- http://www.python.org
- http://www.freeos.com/guides/lsst

C/C++

- http://gcc.gnu.org/onlinedocs/gcc
- http://en.wikipedia.org/wiki/GNU_Compiler_Collection#External_links

Java

- http://www.oracle.com/technetwork/java/index.html
- http://csdl.ics.hawaii.edu/~johnson/613f99/modules/04/jar-files.html
- http://java.sun.com/j2se/1.3/docs/tooldocs/solaris/jdb.html

Linux kernel development

- http://www.kernel.org/pub/linux/docs/lkml/#blkd

Web development

- http://www.onlamp.com
- http://cgi.resourceindex.com

Help with vi

- http://www.freeos.com/guides/lsst/misc.htm#commonvi

19

Installing Linux with AutoYaST2

AutoYaST2 provides the means to install SUSE Linux Enterprise Server without user interaction. With AutoYaST2, a number of installation decisions can be done before the actual installation is started. The YaST installer still checks the hardware and makes decisions accordingly if some values are not set in the AutoYaST2 configuration file.

Other than during cloning, a full installation is done by AutoYaST2. This also means that installing with AutoYaST2 is smaller, but more versatile in its possibilities. The manual interaction of the installation can be done beforehand, and even reused for different installations.

The Linux administration system is now configured as an installation server using NFS to share the installation tree. Configure it as a AutoYaST2 server to perform automated installations over the network.

19.1 Getting started with AutoYaST2

With AutoYaST2, the installation is controlled by information stored in an XML file. This file is created beforehand either manually, or on an existing SUSE Linux Enterprise Server machine with the help of YaST. All of the allowed elements of this XML file are defined in a schema, which can be found in the `/usr/share/YaST2/schema/autoyast/rnc` file after the package `autoyast2` has been installed.

Using AutoYaST2 requires the following steps:

1. Prepare a directory on member 1 LNXADMIN to provide the XML file to the network.
2. Use YaST to create an AutoYaST2 profile.
3. Write a parameter file (parmfile) that accesses the AutoYaST2 XML file on LNXADMIN.
4. Run the automated installation.

19.2 Prepare a directory on LNXADMIN

The Linux administration server exports the /srv/ftp/ directory over NFS. The location of AutoYaST2 profiles will be below /srv/ftp/autoyast/. All the AutoYaST2 profiles will have the naming schema userid.xml:

1. Start an SSH session to LNXADMIN on the single system image (SSI) member with SUSE Linux Enterprise Server 11 SP3.
2. Create a new directory /srv/ftp/autoyast/ with the **mkdir** command:

```
# mkdir /srv/ftp/autoyast
```

The next step is to create an AutoYaST2 profile.

19.3 Setting up an AutoYaST2 profile for LINUX5

The setup of the AutoYaST2 profile will be done with the help of YaST. The preferences for all the settings are taken from the currently running system. More advanced users might want to write the configuration manually to have even more control:

1. Invoke the **yast** command:

```
# yast
```

2. On the left side of the main panel, select **Miscellaneous** and on the right side select **Autoinstallation** and press **Enter**.

   The following window is displayed:

```
[Fileâ][Viewâ][Classesâ][Toolsâ]
 Autoinstallation - Configuration
 +Groups-----------------++Modules----------------+Details
 ¦Hardware               ¦¦General Options        ¦+-----------------------+
 ¦High Availability      ¦¦Power Management       ¦¦Mouse                  -
 ¦Miscellaneous          ¦¦Reporting & Logging    ¦¦                       ¦
 ¦Network Services       ¦¦System Restoration     ¦¦probe                  ¦
 ¦Network Devices        ¦¦System Services (Runlev¦¦                       -
 ¦Security and Users     ¦¦/etc/sysconfig Editor  ¦¦Confirm installation?  ¦
 ¦Software               ¦¦Date and Time          ¦¦                       ¦
 ¦Support                ¦¦Language               ¦¦Yes                    ¦
 ¦System                 ¦¦                       ¦¦                       ¦
 ¦                       ¦¦                       ¦¦Second Stage of        ¦
 ¦                       ¦¦                       ¦¦AutoYaST               ¦
 ¦                       ¦¦                       ¦¦                       ¦
 ¦                       ¦¦                       ¦¦Yes                    ¦
 ¦                       ¦¦                       ¦+-----------------------+
 ¦                       ¦¦                       ¦    [Clone]     [Edit]
 +-----------------------+++----------------¦----+[Apply to system][Clear]
```

Configuring the software

To begin the configuration for the software packages that AutoYaST will install, perform the following steps:

1. On the left side, select **Software** and press **Enter**.
2. On the right side, with the arrow keys select **Package Selection**.
3. With the tab key, move to **Edit** and select it.
4. Move the cursor to **Filter**, use the arrow keys to move to **Patterns**, and press **Enter**.

5. On the left side, move the cursor to **Base System** and press the **space bar** to select it. You should see that *Minimal System* is automatically selected.
6. With the Tab key, move the cursor and select **Accept**.
7. You should see a new panel of more *Automatic Changes*. Select **OK**.

The initial configuration for AutoYaST should now be complete.

Setting the password for root

To set the root password for automatically generated systems, perform the following steps:

1. On the left side of the *Autoinstallation-Configuration* panel, select **Security and Users** and press **Enter**.
2. On the right side select **User and Group Management** and press **Enter**.
3. Move the cursor with the tab key and select **Edit**.
4. Scroll down with the arrow keys to the **root** user and press **Enter**.
5. In the *User Data-Details-Password Settings* panel, change the password in both password fields and use the Tab key to select **OK**.
6. Back in the *User and Group Administration* panel select **OK**.

You should now have the root password set for automatically generated systems.

Configure the host name

To configure the host name for all interfaces, perform the following steps:

1. On the left side of the *Autoinstallation-Configuration* panel, select **Network Devices** and press **Enter**. On the right side, *Network Settings* should be highlighted.
2. Move the cursor to **Edit** and press Enter.
3. In the *Network Settings* panel, move the cursor to top line and use the arrow keys to move to the **Hostname/DNS** tab.
4. Set the *Hostname* field to the correct value. In this example, it is ***virtcook5***:

```
Network Settings
  +Global Options--Overview--Hostname/DNS--Routing---------------------------+
  ¦+Hostname and Domain Name-----------------------------------------------+¦
  ¦¦Hostname                          Domain Name                          ¦¦
  ¦¦virtcook5                                                              ¦¦
  ¦¦[ ] Change Hostname via DHCPNo interface with dhcp                     ¦¦
  ¦¦[ ] Assign Hostname to Loopback IP                                     ¦¦
  ¦+-----------------------------------------------------------------------+¦
  ¦Modify DNS configuration Custom Policy Rule                              ¦
  ¦Only Manually                                   ¦
  ¦+Name Servers and Domain Search List-------------------------------------+¦
  ¦¦Name Server 1                      +Domain Search--------------------+ ¦¦
  ¦¦__________________________________ ¦                                 ¦ ¦¦
  ¦¦Name Server 2                      ¦                                 ¦ ¦¦
  ¦¦__________________________________ ¦                                 ¦ ¦¦
  ¦¦Name Server 3                      ¦                                 ¦ ¦¦
  ¦¦__________________________________ +---------------------------------+ ¦¦
  ¦+-----------------------------------------------------------------------+¦
  +--------------------------------------------------------------------------+
```

The global host name should now be set correctly.

Configure the network interface

To configure the network interface for automatically generated systems, perform the following steps:

1. On the left side of the *Autoinstallation-Configuration* panel, select **Network Devices** and press **Enter**.
2. On the right side, select *Network Settings* should be selected. Move the cursor to **Clone** and press **Enter**. Details should be shown in the rightmost panel.
3. Move the cursor and select **Edit**.
4. The Network Settings panel should appear. Move the cursor to the main window and use the arrow keys to select the **eth0** network card (eth0). In this example, it is `0.0.0600`.
5. Move to **Edit** and press **Enter**.
6. Change the *IP Address* and *Hostname* values to the values correct for your environment. In this example, the IP address is ***9.12.7.5***, the Subnet Mask is ***255.255.240.0***, which translates to a **/20** subnet mask, and the host name is ***virtcook5***:

```
Network Card Setup
  ·General··Address··Hardware·············································
  · Device Type                 Configuration Name                         ·
  · QETH________________________ eth0__________________________            ·
  ·( ) No Link and IP Setup (Bonding Slaves)                               ·
  ·( ) Dynamic Address                                                     ·
  ·                   DHCP__________  DHCP both version 4 and 6_           ·
  ·(x) Statically assigned IP Address                                      ·
  ·IP Address           Subnet Mask           Hostname                     ·
  ·9.12.7.89____________ /22____________ <virtcook5.itso.ibm.com           ·
  ··Additional Addresses··················································
  ··   ·························································   ··
  ··   ·Alias Name·IP Address·Netmask                                ·   ··
  ··   ·                                                             ·   ··
  ··   ·                                                             ·   ··
  ··   ·························································   ··
  ··   [Add][Edit][Delete]                                               ··
  ························································································
  ························································································
 [Help]                 [Back]                 [Cancel]                [Next]
```

7. Move the cursor and select **Next**. You should see the new IP address in the *Network Settings* panel.
8. Move the cursor and select **OK**. This will return you to the main *Autoinstallation-Configuration* panel.

AutoYaST is now configured to define specific network values.

Remove confirmation panel of the installation

By default, the installation will still ask for confirmation before anything is written to disk. This involves manual interaction and might be cumbersome.

To turn off the installation confirmation panel, perform the following steps:

1. On the left side of the *Autoinstallation-Configuration* panel, select **System** and press **Enter**.
2. On the right side, **General Options** should be selected. Move the cursor and select **Edit**.
3. On the *Mouse configuration* panel, accept defaults and select **Next**.

4. On the *Other Options* panel, use the space bar to clear **Confirm installation**, and select **Next**.
5. On the *Ask Options* panel, press **F10** to finish.

With this setting, confirmation before writing to disk will no longer be required.

Save the configuration to disk

Now that the previous settings are complete, you can save the configuration to a file on disk. To do so, perform the following steps:

1. On the *Autoinstallation-Configuration* panel, the **File** menu should be highlighted. Press **Enter**.
2. Use the arrow keys to move down and select **Save As**.
3. Change the directory to /srv/ftp/autoyast/.
4. Move the cursor to the *File name* field and set it to an appropriate name. In this example, **linux5.xml** is used.
5. Move the cursor and select **OK**.
6. The cloning process might take a few seconds. After the cloning has finished, you should see a message similar to "*File /srv/ftp/autoyast/linux5.xml was saved successfully.*". Select **OK**.
7. The **File** menu should be highlighted. Press **Enter** and use the arrow keys to select **Exit.**
8. At the main *YaST2 Control Center* panel, select **Quit** to exit YaST.

The XML file should now be saved in the /srv/ftp/autoyast/ directory. You can use the more command to review the newly created XML file:

```
# more /srv/ftp/autoyast/linux5.xml
<?xml version="1.0"?>
<!DOCTYPE profile>
<profile xmlns="http://www.suse.com/1.0/yast2ns" xmlns:config="http://www.suse.com/1.
0/configns">
  <audit-laf>
    <auditd>
      <action_mail_acct>root</action_mail_acct>
...
```

Add several disks to the system

Adding disks to the system is a twofold process. First, the devices must be activated, and then the partition layout and file systems must be defined. To activate the needed DASD devices, add the following to the XML file inside the <profile>...</profile> tags:

```
<dasd>
    <devices config:type="list">
      <listentry>
        <device>DASD</device>
        <dev_name>/dev/dasda</dev_name>
        <channel>0.0.0100</channel>
        <diag config:type="boolean">false</diag>
      </listentry>
      <listentry>
        <device>DASD</device>
        <dev_name>/dev/dasdb</dev_name>
        <channel>0.0.0300</channel>
        <diag config:type="boolean">false</diag>
      </listentry>
```

```
      <listentry>
        <device>DASD</device>
        <dev_name>/dev/dasdc</dev_name>
        <channel>0.0.0301</channel>
        <diag config:type="boolean">false</diag>
      </listentry>
    </devices>
  </dasd>
```

These statements add the devices 0100, 0300, and 0301 to the system. The actual partitioning is done in a separate section, again within the <profile>...</profile> tags:

```
<partitioning config:type="list">
<drive> 1
<device>/dev/dasda</device>
<initialize config:type="boolean">true</initialize> 2
<partitions config:type="list"> 3
<partition>
  <create config:type="boolean">true</create> 4
  <format config:type="boolean">true</format> 5
  <mount>/</mount> 6
  <partition_nr config:type="integer">1</partition_nr> 7
  <size>max</size>
</partition>
</partitions>
    </drive>
    <drive>
      <device>/dev/dasdb</device>
      <use>all</use>8
      <partitions config:type="list">
        <partition>
          <create config:type="boolean">true</create>
          <format config:type="boolean">true</format>
          <filesystem config:type="symbol">swap</filesystem> 9
          <partition_nr>1</partition_nr>
          <mount>swap</mount>
          <fstopt>pri=2</fstopt>
        </partition>
      </partitions>
    </drive>
    <drive>
      <device>/dev/dasdc</device>
      <use>all</use>
      <partitions config:type="list">
        <partition>
          <create config:type="boolean">true</create>
          <format config:type="boolean">true</format>
          <filesystem config:type="symbol">swap</filesystem>
          <partition_nr>1</partition_nr>
          <mount>swap</mount>
          <fstopt>pri=1</fstopt>
        </partition>
      </partitions>
    </drive>
  </partitioning>
```

Notes:

1 All of the respective disks are handled within <drive>...</drive> elements. For each of the disks, one of these sections must be created.

2 If true, *initialize* wipes out all of the partitions on that disk.

3 The *partitions* element is a container for several partitions. In the example described here, only one partition is set up for each disk.

4 After wiping the partitions, AutoYaST2 must create the new partitions before they can be used. The *create* element takes control of this.

5 The new partition will be formatted if *format* is set to true.

6 The mount point can be any directory including / or use swap as a special keyword.

7 Only one partition is created for each disk in this setup. Therefore, the `partition_nr` is set to 1.

8 By using all, any partitions available on the selected disk are used for the new configuration.

9 For swap, the file system type swap must be selected.

This setup re-creates the SWAP disks during the AutoYaST2 installation.

As a reference, the configuration used in this section is supplied with the files associated with this book. These should be available at:

```
ftp://<LNXADMIN.ip.address>/SG248147/sles11sp3/linux5.xml
```

19.4 Running an automated installation

To start the installation, a new parameter file has to be created. To do so, perform the following steps:

1. Log on as `LNXMAINT`.
2. Copy the existing parameter file, `S113GOLD PARM-S11`, to a new file with the user ID as the file name. In this example, it is `LINUX5`:

   ```
   ==> copy s113gold parm-s11 d linux5 = =
   ```

3. Edit the new file and change the values of `HostIP`, `host name`, and `Install` to those that are correct for your environment. Add a new line to reference the AutoYaST2 profile:

   ```
   ==> x linux5 parm-s11 d
   ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dumb
   HostIP=9.12.7.5         Hostname=virtcook5.itso.ibm.com
   Gateway=9.12.4.1        Netmask=255.255.240.0 Layer2=1
   ReadChannel=0.0.0600  WriteChannel=0.0.0601  DataChannel=0.0.0602
   Nameserver=9.12.6.7        portname=whatever portno=0
   Install=ftp://9.12.7.8/SLES OSAHWAddr=
   UseSSH=1 SSHPassword=12345678
   UseVNC=1  VNCPassword=12345678
   InstNetDev=osa OsaInterface=qdio OsaMedium=eth Manual=0
   autoyast=nfs://9.12.7.8/srv/ftp/autoyast/linux5.xml
   ```

4. Save the file.
5. Log off from `LNXMAINT`.
6. Log on to `LINUX5`.
7. Start the installation with the install EXEC. The autoyast line will be read, and the xml file will be used for an automated installation. Watch for the new **autoyast** line in the parameter file:

   ```
   ==> sles11s3
   Initializing cgroup subsys cpuset
   Initializing cgroup subsys cpu
   ```

```
Linux version 3.0.76-0.9-default (geeko@buildhost) (gcc version 4.3.4 [gcc-4_3-b
ranch revision 152973] (SUSE Linux) ) #1 SMP Fri May 31 09:17:47 UTC 2013 (82ad5
ef)
setup.1a06a7: Linux is running as a z/VM guest operating system in 64-bit mode
Zone PFN ranges:
  DMA      0x00000000 -> 0x00080000
  Normal   empty
Movable zone start PFN for each node
early_node_map[1] active PFN ranges
    0: 0x00000000 -> 0x00020000
PERCPU: Embedded 10 pages/cpu @0000000002242000 s11776 r8192 d20992 u40960
Built 1 zonelists in Zone order, mobility grouping on.  Total pages: 129280
Kernel command line: ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dum
b                    HostIP=9.12.7.5        Hostname=virtcook5.itso.ibm.com
                     Gateway=9.12.4.1       Netmask=255.255.240.0 Layer2=1
                     ReadChannel=0.0.0600  WriteChannel=0.0.0601  DataChannel=0.
0.0602               Nameserver=9.12.6.7       portname=whatever portno=0
                     Install=ftp://9.12.7.8/SLES
                     UseSSH=1 SSHPassword=12345678
                     UseVNC=1  VNCPassword=12345678
                     InstNetDev=osa OsaInterface=qdio OsaMedium=eth Manual=0
                     autoyast=nfs://9.12.7.8/srv/ftp/autoyast/linux5.xml
```

8. You will likely be prompted for a MAC address and DNS name again. Press **Enter** twice for each prompt:

```
...
MAC address
>
...
Enter the IP address of your name server. Leave empty or enter "+++" if you
don't need one
[9.12.6.7]>
...
starting VNC server...
A log file will be written to: /var/log/YaST2/vncserver.log ...

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

(When YaST2 is finished, close your VNC viewer and return to this window.)

Active interfaces:

eth0      Link encap:Ethernet  HWaddr 02:00:0C:00:00:28
          inet addr:9.12.7.5  Bcast:9.12.15.255  Mask:255.255.240.0
--
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0

YaST2.call (2255): /proc/2846/oom_adj is deprecated, please use /proc/2846/oom_s
core_adj instead.
*** Starting YaST2 ***
md: raid0 personality registered for level 0
md: raid1 personality registered for level 1
raid6: int64x1   2423 MB/s
raid6: int64x2   4059 MB/s
```

Now the entire installation process should be totally automated.

9. The installation takes quite a while to complete. Be patient.
10. When the installation process is complete, you should be able to log in to your new virtual server.

Important: During the writing of this chapter, the host name of the newly cloned server remained that of the golden image. After the **dns** section in the XML file below was deleted, the correct host name was assigned:

```
...
<networking>
    <dns>
      <dhcp_hostname config:type="boolean">false</dhcp_hostname>
      <domain>itso.ibm.com</domain>
      <hostname>virtcook8</hostname>
      <nameservers config:type="list">
        <nameserver>9.12.6.7</nameserver>
      </nameservers>
      <resolv_conf_policy>auto</resolv_conf_policy>
      <searchlist config:type="list">
        <search>itso.ibm.com</search>
      </searchlist>
      <write_hostname config:type="boolean">false</write_hostname>
    </dns>
  ...
```

20

Creating appliances with KIWI

KIWI provides a mechanism to create images of SUSE Linux Enterprise Servers. These can be installation DVD images, live images that can run without deployment to disk, preloaded operating systems, or readily configured appliances.

KIWI is closer to cloning an existing system, while AutoYaST is an automated installation. However, unlike cloning, KIWI uses configuration files and repositories and the actual imaging process does not need any further manual interaction. One important difference from cloning is that it is possible to create images that never have been booted. So KIWI is somewhere between cloning and automated building of a system.

Images that never have been booted, and doing the final configuration during the first boot, are also called *preload images* or *firstboot images*. The big advantage of this is that all unique strings and files inside an operating system are commonly generated when a service is started up the first time. All the unique strings and files must be removed from a golden image, which can be a problem if you do not know all of the possible locations. On a preload image, these unique strings and files have never been generated, and thus it is simply not necessary to know about the locations.

Compared to automated installations, starting a preconfigured image is much faster and also uses a lot less hardware resources.

Preparing such an image is very easy for small images with only little functionality. Preparing an image with complex functionality can become a time-consuming process. As a rule of thumb, it is worthwhile to create an image when one of the following statements is true:

- The same image is used multiple times.
- The image should be always available with the latest updates.
- It must be possible to reproduce the image from a set of rules.

A good candidate for an image is a very small version of the operating system that is used as a minimal configuration for all servers. This is also known as *JeOS*. The following requirements apply for this system:

- It uses only standard packages.
- The software update and installation stack must be fully functional (zypper).
- It is directly bootable.

The network is either updated from a parameter file as described in the Chapter 17, “Clone SUSE Linux Enterprise Server 11 SP3” on page 305, or by configuring DHCP to be used.

There is a development front end to KIWI available, which is called *SUSE Studio*. However, this is not freely available but a priced feature. Besides providing a nice front end, SUSE Studio also adds a number of development features, like checking for differences between versions of images or inheritance of other images. For x86, this front end is publicly available on the Internet at http://susestudio.com on condition that only open source packages are used. For mainframe environments, SUSE Studio is a priced feature that must be deployed in a local environment. This chapter goes forward only with the methods and features available from the SDK of SLE11SP3.

> **Note:** When running in an environment where many rebuilds of packages are expected and some automated deployment process must be implemented for those packages, it could be worth while to take a closer look at the *Open Build Service* (OBS).
>
> The OBS is providing a sophisticated infrastructure to create packages for different architectures and different distributions. It is available as open source software, and used by SUSE to run all needed build jobs. Some of its main features are:
>
> - Version control of all software packages.
> - Support for multiple architectures.
> - Support for different package formats like rpm, and deb, as well as image builds.
> - Automated resolving of package dependencies.
> - Automatic rebuild of all depending packages if binary changes occur.
> - Publishing to repositories that may be registered by zypper.
> - Code review before check in to a repository.
> - And many more.
>
> All of the features can publicly be used for the openSUSE project by using the command line tool **`osc`**. A tutorial for this is available at the following site:
> http://en.opensuse.org/openSUSE:Build_Service_Tutorial.
>
> The central documentation page with references on how to deploy the OBS locally is available at http://en.opensuse.org/Portal:Build_Service.

20.1 Set up KIWI on LNXADMIN

KIWI needs some preparation steps before it can be used. For example, additional disk space is needed to store images.

20.1.1 Add disks for KIWI

Two extra minidisks are added to `LNXADMIN` on the second single system image (SSI) member for storing images created by KIWI. The root file system will be kept in `/var/tmp/kiwi/` adhering to the Linux File Hierarchy Standard (FHS).

Table 20-1 Disks used for KIWI

Minidisk	Cylinders	Mount point	Usage
`200`	3338	NONE	Imaging disk
`201`	10016	`/var/tmp/kiwi/`	Preparation of root file system

Add two lines to the appropriate SUBCONFIG section of the IDENTITY LNXADMIN. In this example the disks have labels of ***JM113A*** and ***JM1364***:

```
SUBCONFIG LNXADM-2
 MDISK 0100 3390 0001 10016 JM1362 MR READPASS WRITPASS MULTPASS
 MDISK 0101 3390 0001 10016 JM1363 MR READPASS WRITPASS MULTPASS
 MDISK 0200 3390 0001  3338 JM113A MR READPASS WRITPASS MULTPASS
 MDISK 0201 3390 0001 10016 JM1364 MR READPASS WRITPASS MULTPASS
```

Set up the 200 minidisk

The 200 minidisk does not need a file system. However, it must be preformatted with **dasdfmt**:

- Start an SSH session as root to the Linux running on LNXADMIN.
- Run the **dasd_configure** command to make the disk available across reboots:

```
# dasd_configure 0.0.0200 1 0
Device 0.0.0200 is unformatted
```

- Check the resulting device letter with the **lsdasd** command:

```
# lsdasd
Bus-ID     Status      Name      Device  Type  BlkSz  Size      Blocks
==========================================================================
0.0.0100   active      dasda     94:0    ECKD  4096   7042MB    1802880
0.0.0301   active      dasdb     94:4    FBA   512    512MB     1048576
0.0.0300   active      dasdc     94:8    FBA   512    256MB     524288
0.0.0101   active      dasdd     94:12   ECKD  4096   7042MB    1802880
0.0.0200   n/f         dasde     94:20   ECKD
```

In this example, the device /dev/dasde is the newly created device, which now has to be formatted.

- Use the **dasdfmt** command to format the disk. In this example, it is /dev/dasde:

```
# dasdfmt -b 4096 -f /dev/dasde
Drive Geometry: 3338 Cylinders * 15 Heads =  50070 Tracks

I am going to format the device /dev/dasde in the following way:
   Device number of device : 0x200
   Labelling device        : yes
   Disk label              : VOL1
   Disk identifier         : 0X0200
   Extent start (trk no)   : 0
   Extent end (trk no)     : 50069
   Compatible Disk Layout  : yes
   Blocksize               : 4096

--->> ATTENTION! <<---
All data of that device will be lost.
Type "yes" to continue, no will leave the disk untouched: yes
```

This command will take some time.

- Use the **fdasd** command to create a single partition on this disk:

```
# fdasd -a /dev/dasde
reading volume label ..: VOL1
reading vtoc ..........: ok

auto-creating one partition for the whole disk...
writing volume label...
writing VTOC...
rereading partition table...
```

The new partition, /dev/dasde1, is ready to use as an imaging disk.

Set up the 201 minidisk

To configure the 201 minidisk, a new volume group will be created and mounted over /var/tmp/kiwi/:

- Run the **dasd_configure** command to make the disk available across reboots:

```
# dasd_configure 0.0.0201 1 0
Device 0.0.0201 is unformatted
```

- Check the resulting device letter with the **lsdasd** command:

```
# lsdasd
Bus-ID     Status      Name      Device  Type  BlkSz  Size      Blocks
==============================================================================
0.0.0301   active      dasda     94:0    FBA   512    513MB     1050624
0.0.0100   active      dasdb     94:4    ECKD  4096   7042MB    1802880
0.0.0101   active      dasdc     94:8    ECKD  4096   7042MB    1802880
0.0.0300   active      dasdd     94:12   FBA   512    257MB     526336
0.0.0200   active      dasde     94:16   ECKD  4096   2347MB    600840
0.0.0201   active      dasdf     94:20   ECKD  4096   7042MB    1802880
```

- Use the **dasdfmt** command to format the disk. In this example, it is /dev/dasdf:

```
# dasdfmt -b 4096 -f /dev/dasdf
Drive Geometry: 10016 Cylinders * 15 Heads =  150240 Tracks

I am going to format the device /dev/dasdf in the following way:
   Device number of device : 0x201
   Labelling device        : yes
   Disk label              : VOL1
   Disk identifier         : 0X0201
   Extent start (trk no)   : 0
   Extent end (trk no)     : 150239
   Compatible Disk Layout  : yes
   Blocksize               : 4096

--->> ATTENTION! <<---
All data of that device will be lost.
Type "yes" to continue, no will leave the disk untouched: yes
```

- Use the **fdasd** command to create a single partition on this disk:

```
# fdasd -a /dev/dasdf
reading volume label ..: VOL1
reading vtoc ..........: ok

auto-creating one partition for the whole disk...
writing volume label...
writing VTOC...
rereading partition table...
```

- Create a physical volume from the partition with the **pvcreate** command:

```
# pvcreate /dev/dasdf1
  Physical volume "/dev/dasdf1" successfully created
```

- Create a volume group named **var-tmp-vg** with the **vgcreate** command:

```
# vgcreate var-tmp-vg /dev/dasdf1
Volume group "var-tmp-vg" successfully created
```

- Observe the number of extents in the volume group with the **vgdisplay** command:

```
# vgdisplay var-tmp-vg
```

```
--- Volume group ---
VG Name               var-tmp-vg
System ID
Format                lvm2
Metadata Areas        1
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                1
Act PV                1
VG Size               6.88 GiB
PE Size               4.00 MiB
Total PE              1760
Alloc PE / Size       0 / 0
Free  PE / Size       1760 / 6.88 GiB
VG UUID               wbCi1W-y4zr-ShIF-FK9r-fk9K-OOO5-I5FjPT
```

In this example, there are 1760 extents free.

- Create a logical volume named **kiwi** with the **lvcreate** command using all extents:

```
# lvcreate -l 1760 -n kiwi var-tmp-vg
Logical volume "kiwi" created
```

- Create an XFS file system with the **mkfs** command:

```
# mkfs -t xfs /dev/var-tmp-vg/kiwi
meta-data=/dev/var-tmp-vg/kiwi   isize=256    agcount=8, agsize=225280 blks
         =                       sectsz=4096  attr=2, projid32bit=0
data     =                       bsize=4096   blocks=1802240, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0
log      =internal log           bsize=4096   blocks=2560, version=2
         =                       sectsz=4096  sunit=1 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
```

- Make a backup copy of the working `/etc/fstab` file:

```
# cd /etc
# cp fstab fstab.works
```

- Edit the `fstab` file and add a line so that the new logical volume will be mounted over the `/var/tmp/kiwi/` directory:

```
# vi fstab
/dev/disk/by-path/ccw-0.0.0300-part1 swap                swap       pri=2
0 0
/dev/disk/by-path/ccw-0.0.0301-part1 swap                swap       pri=1
0 0
/dev/disk/by-path/ccw-0.0.0100-part1 /                  ext3       acl,user_xattr
1 1
/dev/lnxadmin-vg/srv /srv                xfs        defaults              1 2
/dev/var-tmp-vg/kiwi /var/tmp/kiwi       xfs        defaults              0 0
...
```

- Make the `/var/tmp/kiwi/` directory for the mount point:

```
# mkdir /var/tmp/kiwi
```

- Use the **mount -a** command to mount all file systems. This has the side effect of testing the syntax in the `/etc/fstab` file:

```
# mount -a
```

- Verify that the new logical volume is mounted:

```
# mount | grep kiwi
/dev/mapper/var--tmp--vg-kiwi on /var/tmp/kiwi type xfs (rw)
```

A new logical volume has been created by using the `201` minidisk and will be mounted over `/var/tmp/kiwi/` at boot time.

20.1.2 Install the required packages

A number of packages are needed to run KIWI on LNXADMIN. Some of these are only available from the SDK:

- Start an SSH session to `LNXADMIN` as root.
- Ensure that you have the SDK available with the following **zypper** command:

```
# zypper lr -d
# | Alias | Name...
--+--------------------------------------------------------------------+-----...
1 | SUSE-Linux-Enterprise-Server-11-SP3_11.3.3-1.138                   |SUSE-...
2 | SUSE-Linux-Enterprise-Software-Development-Kit-11-SP3_11.3.3-1.69 |SUSE-...
```

  The lines have been shortened for better visibility. If you do not have the SDK available, add it with yast → Software → Software Repositories.

- Install the RPMs `kiwi` and `kiwi-desc-oemboot`. A number of dependencies will be automatically resolved, which leads to installation of several more packages:

```
# zypper in kiwi kiwi-desc-oemboot squashfs
Loading repository data...
Reading installed packages...
Resolving package dependencies...

The following NEW packages are going to be installed:
  compat java-1_4_2-ibm jing jpackage-utils kiwi kiwi-desc-oemboot kiwi-tools
  perl-HTML-Parser perl-HTML-Tagset perl-IO-Socket-INET6 perl-IO-Socket-SSL
  perl-Net-SSLeay perl-Socket6 perl-libwww-perl virt-utils squashfs

The following recommended packages were automatically selected:
  jing perl-IO-Socket-INET6 perl-IO-Socket-SSL

The following packages are not supported by their vendor:
  java-1_4_2-ibm jing kiwi kiwi-desc-oemboot kiwi-tools

16 new packages to install.
Overall download size: 39.5 MiB. After the operation, additional 82.5 MiB will
be used.
Continue? [y/n/?] (y): y
```

- Answer with **y**. The packages will now be added to the system.

KIWI and all related packages should now be installed.

20.1.3 Create a basic file system structure

When KIWI runs, it expects all the configurations below a directory that is given by the command line. Inside this directory, there is the `config.xml` as well as a file named `.checksum.md5`. In addition to this, a root file system with overlay files can be added to this directory:

- Start an SSH session to `LNXADMIN` as `root`.

- Create the /root/kiwi/template/root/ directory with the following **mkdir** command:

```
# mkdir -p /root/kiwi/template/root
```

- Create a basic KIWI configuration file with the following content:

```
# cd /root/kiwi/template
# vi config.xml
<?xml version="1.0" encoding="utf-8"?> 1

<image schemaversion="5.2" name="SLES-11-SP3-preload"> 2
  <description type="system"> 3
    <author>BGunreben</author>
    <contact>bg@example.com</contact>
    <specification>SLES 11 SP3 preload system</specification>
  </description>
  <preferences> 4
    <type image="oem"
          filesystem="ext3"
          boot="oemboot/suse-SLES11"
          bootloader="zipl"
          installiso="false">
      <oemconfig>
        <oem-boot-title>Example Preload System</oem-boot-title>
      </oemconfig>
    </type>
    <version>1.0.0</version>
    <packagemanager>zypper</packagemanager>
    <rpm-check-signatures>false</rpm-check-signatures>
    <rpm-force>true</rpm-force>
    <locale>en_US</locale>
    <keytable>us.map.gz</keytable>
  </preferences>
  <users group="root">
    <user pwd="" home="/root" name="root"/>
  </users>
  <repository type="yast2"> 5
    <source path="dir:///srv/ftp/SLES-11-SP3"/>
  </repository>
  <repository type="yast2">
    <source path="dir:///srv/ftp/SLE-11-SP3-SDK"/>
  </repository>
  <packages type="image" patternType="onlyRequired"> 6
    <opensusePattern name="base"/>
    <opensusePattern name="minimal"/>
  </packages>
  <packages type="bootstrap">
    <package name="filesystem"/>
    <package name="glibc-locale"/>
  </packages>
</image>
```

Notes:

1 This file is defined as an XML file, following XML version 1.0. The allowed character set is UTF-8.

2 All KIWI image configurations have **image** as root element. For those that are familiar with XML schemas, the definition for this schema can be found in the /usr/share/kiwi/module directory.

3 Here, add yourself to the description. The specification will likely be changed for each new appliance that is created.

4 In the **preferences** element a number of basic decisions are made. Especially the **type** element has a major effect on the resulting image. For mainframe, the most useful type is **oem**. With this type, a preload image is created which can easily be deployed to other disks by simply dumping it on a device.

5 To make KIWI work correctly, at least two repositories are needed. One is the SLES installation repository, and the other is the SDK. Without the SDK, kiwi would not be able to find all the needed packages during the image build. Here, it is also possible to add several more repositories. If the build systems have been set up correctly, all dependencies will be resolved automatically.

6 The packages that are installed can be complete patterns, or just packages. Possible dependencies will automatically be resolved. Therefore, the number of packages that is actually installed might be higher than expected.

- After creating the configuration file, check that the XML is well formatted:

```
# xmlwf config.xml
```

- Do a first run of the KIWI preparation step. First, create the checksum file of config.xml. This has to be renewed after any change of the configuration:

```
# md5sum config.xml > .checksum.md5
```

- Run the **kiwi** command as follows:

```
# cd /root/kiwi
# kiwi --prepare template --root /var/tmp/kiwi/img_root
Jul-26 15:17:46 <1> : Reading image description [Prepare]...
...
Jul-26 15:18:18 <1> : KIWI exited successfully                       done
Jul-26 15:18:18 <1> : Complete logfile at: /var/tmp/kiwi/img_root.log     done
```

- If you want to reuse the same image root directory, you will have to delete the current root directory first. However, if you want to create a real image from that directory, you must not touch that directory. If you really want to remove that directory, the command `rm -rf /var/tmp/kiwi/img_root` will accomplish that.

20.2 Creating a minimal operating system

The starting point for configuring most servers is a minimal image, that has just the needed functionality to connect to it, and to install software. SUSE also calls such an operating system a *JeOS*.

Deployment of such an image as standard procedure provides a means to have a consistent setup for all minimal systems. It is also possible to have some standard login mechanism or security measure that is preinstalled for every machine that way.

20.2.1 Create a directory structure for the new project

When starting a new project, it is a good practice to create a file system structure. To speed up the start, it is possible to simply copy the template directory to a new directory:

- This can also be done with following **kiwi** command:

```
# cd /root/kiwi
# kiwi --clone template -d jeos
Jun-19 13:23:01 <1> : Cloning image template -> /root/kiwi/jeos...
Jun-19 13:23:01 <1> : Destination: /root/kiwi/jeos doesn't exist
Jun-19 13:23:01 <1> : Would you like kiwi to create it [yes/no] ?
```

```
Jun-19 13:23:01 <1> : Would you like kiwi to create it [yes/no] ? yes
Jun-19 13:23:04 <1> : KIWI exited successfully                        done
```

- Show the contents of the new directory:

```
# ls -l jeos
-rw-r--r-- 1 root root 1295 Jul 26 15:13 config.xml
drwxr-xr-x 2 root root 4096 Jul 26 14:47 root
```

20.2.2 Select packages to install

The selection of packages can be a time consuming task. The exact selection of packages also may vary between different deployments. After getting started with KIWI in section 20.1, “Set up KIWI on LNXADMIN” on page 350, there is already a system available that used the two patterns “base” and “minimal”. To create a list of needed packages in a format that can be reused by KIWI, proceed as follows:

- Edit the `jeos/config.xml` file and replace the following lines with your package list:

```
<opensusePattern name="base"/>
<opensusePattern name="minimal"/>
```

  Also, remove the attribute **patternType="onlyRequired"** from the surrounding packages element.

- Make sure to add the package **cmsfs** to the list.

20.2.3 Create the needed overlay files

A number of files should be replaced in the image to prepare the correct configuration.

Prepare the image for cloning

The script boot.clone should be run during the first boot. Doing this will configure the network according to the prepared parameter file on the 191 disk:

```
# cd /root/kiwi/jeos
# mkdir -p root/etc/init.d
# cp -a /srv/ftp/SG248147/sles11sp3/boot.clone root/etc/init.d/
```

Prepare the OSA interface

All of the virtual machines do have the same virtual OSA devices. This makes configuration somewhat easy. Copy the needed files from LNXADMIN to the correct directory:

```
# cd /root/kiwi/jeos/root
# mkdir -p etc/udev/rules.d
# cp -a /etc/udev/rules.d/51-qeth-0.0.0600.rules etc/udev/rules.d
# cp -a /etc/udev/rules.d/70-persistent-net.rules etc/udev/rules.d
# cp -a /etc/udev/rules.d/77-network.rules etc/udev/rules.d
```

Prepare the system to access LNXADMIN as installation server

The system is very minimal, and it is very likely, that the admin will need to install several packages after the first bootup. This is very easy if the system is prepared for this:

```
# cd /root/kiwi/jeos/root
# mkdir -p etc/zypp/repos.d
# vi SUSE-Linux-Enterprise-Server-11-SP3_11.3.3-1.138_1.repo
[SUSE-Linux-Enterprise-Server-11-SP3_11.3.3-1.138_1]
name=SUSE-Linux-Enterprise-Server-11-SP3 11.3.3-1.138
enabled=1
autorefresh=1
```

```
baseurl=ftp://9.12.7.8/SLES-11-SP3
path=/
type=yast2
keeppackages=0
```

Also prepare the SDK installation repository:

```
# vi SUSE-Linux-Enterprise-Software-Development-Kit-11-SP3_11.3.3-1.69.repo
[SUSE-Linux-Enterprise-Software-Development-Kit-11-SP3_11.3.3-1.69]
name=SUSE-Linux-Enterprise-Software-Development-Kit-11-SP3 11.3.3-1.69
enabled=1
autorefresh=0
baseurl=ftp://9.12.7.8/SLE-11-SP3-SDK
path=/
type=yast2
keeppackages=0
```

20.2.4 Write a config.sh script

If available, a script called **config.sh** is called in the readily prepared directory structure for the script. This makes it possible to add special functions like additions to the init process. In the case described here, it is necessary to active the init script **boot.clone**:

```
# cd /root/kiwi/jeos
# vi config.sh
#!/bin/bash

test -f /.kconfig && . /.kconfig
test -f /.profile && . /.profile

suseActivateDefaultServices
suseInsertService boot.clone
exit 0
```

20.2.5 Create and clone the image

As already described in section 20.1, “Set up KIWI on LNXADMIN” on page 350, the kiwi build process consists of two main steps. To keep the image of the template configuration on disk, choose a new destination directory for the unpacked tree:

```
# cd /root/kiwi
# kiwi --prepare jeos --root /var/tmp/kiwi/img_jeos
Jun-19 14:19:41 <1> : Reading image description [Prepare]...
Jun-19 14:19:41 <1> : Set root log: /var/tmp/kiwi/img_jeos.3005.screedonelog...
Jun-19 14:19:41 <1> : Setting up package manager: zypper            done
Jun-19 14:19:41 <1> : Setting RPM doc exclusion to: true            done
Jun-19 14:19:41 <1> : Creating default template files for new root sydone
...
```

During the second step, the image is built, and copied to the 200 disk, where the boot loader is installed. It then is retrieved from the disk and copied to /var/tmp/kiwi/image. The image however is still available on the disk 200, which is disk /dev/dasdf in this example. This makes it easy to just use **FLASHCOPY** to copy the image to the target disk:

- Check which device is your 200 disk:

```
# lsdasd
Bus-ID     Status      Name      Device  Type  BlkSz  Size      Blocks
==============================================================================
0.0.0100   active      dasda     94:0    ECKD  4096   7042MB    1802880
```

```
0.0.0301   active      dasdb     94:4    FBA   512    512MB     1048576
0.0.0101   active      dasdc     94:8    ECKD  4096   7042MB    1802880
0.0.0201   active      dasdd     94:12   ECKD  4096   7042MB    1802880
0.0.0300   active      dasde     94:16   FBA   512    256MB     524288
0.0.0200   active      dasdf     94:20   ECKD  4096   2347MB    600840
```

- Check the size of the disk:

```
# vmcp q v 200
DASD 0200 3390 JM113A R/W       3338 CYL ON DASD  113A SUBCHANNEL = 0002
```

- Create an image from the configuration with the following command:

```
# kiwi -c /var/tmp/kiwi/img_jeos -d /var/tmp/kiwi/image --targetDevice /dev/dasdf
```

  This process might take a while.

- Check that the target system, `LINUX6` in this example, is not logged on:

```
# vmcp q LINUX6
HCPCQU045E LINUX6 not logged on
Error: non-zero CP response for command 'Q LINUX6': #45
```

  This error message shows `LINUX6` is not logged on.

- Link the target disk to the local machine:

```
# vmcp link linux6 100 1100 mr
```

- Check the size of the target disk. This must have at least the 3338 cylinders that our 200 disk had available. If it is bigger, the file system of the JeOS image will be resized during the first bootup:

```
# vmcp q v 1100
DASD 1100 3390 JM126B R/W      10016 CYL ON DASD  126B SUBCHANNEL = 0015
```

- If the target disk has never been used before, use **dasdfmt** to initialize it:

```
# dasd_configure 0.0.1100 1 0
# lsdasd | grep 1100
0.0.1100   active      dasdg     94:24   ECKD  4096   3521MB    901440
# dasdfmt -b 4096 -y -f /dev/dasdg
# dasd_configure 0.0.1100 0 0
```

  This is only needed the first time; this DASD is used by Linux.

- Use `FLASHCOPY` to copy the disk. If it is not available, you can also use **DDR** at the z/VM level, or the manual procedure described in 17.2, “Clone a virtual server manually” on page 306. Only the 3338 cylinders of the `200` device are copied. The last cylinder is 3337 because the first cylinder starts at 0:

```
# vmcp flashcopy 0200 0 end 1100 0 3337
```

- Detach the target disk from the local system:

```
# vmcp detach 1100
```

20.2.6 Start the image

Before starting the image, make sure to have the network information added to the parmfile `LINUX6 PARM-S11` on the `191` disk of `LINUX6`. This can be checked with the following commands:

```
# dasd_configure 0.0.0191 1 0
# lsdasd | grep 191
0.0.0191   active(ro)  dasdg     94:24   ECKD  4096   351MB     90000
# cmsfscat -d /dev/dasdg linux6.parm-s11
ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc
```

```
TERM=dump HostIP=9.12.7.6 Hostname=virtcook6
Gateway=9.12.4.1 Netmask=255.255.240.0 Layer2=1
ReadChannel=0.0.0600  WriteChannel=0.0.0601
DataChannel=0.0.0602 OSAHWAddr=
Nameserver=9.12.6.7 Portname= Portno=0
Install=ftp://9.12.7.8/SLES-11-SP3
UseVNC=1  VNCPassword=12345678
UseSSH=1 SSHPassword=12345678
InstNetDev=osa OsaInterface=qdio OsaMedium=eth Manual=0
```

After doing this, the system is ready for booting:

- Log on as LINUX6.
- Confirm to boot from minidisk 100.
- Wait until the system is up and running. The first time, this will take longer than subsequent times because the file system is resized, and several initialization steps have to be done during the first boot.

20.3 Creating a basic WebSphere instance

When creating appliances for third-party software, it is necessary to have enough knowledge about the normal installation of the respective software. As an example, when getting the software of the IBM WebSphere® Application Server, you might get a file like *wlp-base-trial-runtime-8.5.5.0.jar*. To add this software to a new project proceed as follows:

- Copy the JAR file to the /root/ directory.
- Create a clone of the JeOS image:

  ```
  # cd /root/kiwi
  # kiwi --clone jeos -d wlp
  ```

  A new project is created with wlp as base directory.
- Create a target directory for the software:

  ```
  # mkdir -p wlp/root/opt/IBM
  ```
- Unpack the software to the kiwi project:

  ```
  # java -jar /root/wlp-base-trial-runtime-8.5.5.0.jar
  ```
- Accept the license, and change the destination directory to:

  ```
  # cd /root/kiwi/wlp/root/opt/IBM
  ```
- Edit the kiwi configuration file and add several packages:

  ```
  <package name="java-1_7_0-ibm"/>
  <package name="java-1_7_0-ibm-devel"/>
  <package name="xinetd"/>
  ```
- Create an initialization script that starts the service during startup.
  - The first part is the informing the system about when to start the service in the sequence of several other services:

    ```
    #!/bin/bash
    #
    # /etc/init.d/wlp
    #
    ### BEGIN INIT INFO
    # Provides:          wlp
    # Required-Start:    network
    ```

```
# Required-Stop:     network
# Default-Start:     3
# Default-Stop:
# Short-Description: Start the wlp server
# Description:       Short init script to run this service
### END INIT INFO
```

– Next, is two lines that enable the coloring of status messages:

```
. /etc/rc.status

rc_reset
```

– The actual start, stop, and status functions are done just next:

```
case "$1" in
    start)
        /opt/IBM/wlp/bin/server start
        rc_status
        ;;
    stop|restart)
        /opt/IBM/wlp/bin/server stop
        rc_status
        ;;
    status)
        /opt/IBM/wlp/bin/server status
        rc_status
        ;;
    *)
        echo "Usage: $0 {start}."
        exit 1
        ;;
esac
rc_exit
```

- Before actually building the appliance, some modifications are needed to the file `wlp/config.sh`:

```
# vi wlp/config.sh
#!/bin/bash

test -f /.kconfig && . /.kconfig
test -f /.profile && . /.profile

suseActivateDefaultServices
suseInsertService boot.clone
suseInsertService wlp

exit 0
```

To start the actual build, first remove some eventually old image root directories. Then, do the prepare and build of the appliance:

- Remove the image root of previous build:

```
# rm -rf /var/tmp/kiwi/img_wlp*
```

- Run the preparation of the image:

```
# kiwi --prepare wlp --root /var/tmp/kiwi/img_wlp
```

- Do the actual image build:

```
# kiwi -c /var/tmp/kiwi/img_wlp -d /var/tmp/kiwi/image --targetDevice /dev/dasdf
```

This example supposes that you already did set up /dev/dasdf as special device that can be used and erased during the build.

- Make sure that the parameter file for the target virtual machine is prepared. In this example `LINUX6`.
- Make sure that the new virtual machine is logged off.
- Copy the image to the new disks with the following commands:

```
# vmcp link linux6 100 1100 mr
# vmcp flashcopy 0200 0 end 1100 0 3337
# vmcp det 1100
```

- Log on the new guest. Wait until it is fully up, then check your configuration by pointing your browser to the new WebSphere instance.

> **Note:** If you do not have access to the needed ports of the target machine, or if you want to do as though you are local on the target machine, you might want to use the xinetd port forwarding feature. To do this, create a file `/etc/xinetd.d/wlp`:
>
> ```
> # vi /etc/xinetd.d/wlp
> service was-http
> {
>         disable                 = no
>         type                    = UNLISTED
>         socket_type             = stream
>         protocol                = tcp
>         wait                    = no
>         redirect                = localhost 9080
>         port                    = 80
>         user                    = nobody
> }
> ```
>
> After starting xinetd on the target machine, point your local browser to the target machine at standard http port 80. This will redirect the network packages to localhost at port 9080.
>
> This feature is probably not appropriate for production, and it will not take the load of a fully operational server, but it is nice for development.

20.4 More information

KIWI is a complex imaging system. There is quite an amount of documentation available to explain this topic.

The following sources provide good entry points:

- http://en.opensuse.org/Portal:KIWI
- A cookbook is available at the following site:

  http://doc.opensuse.org/projects/kiwi/doc
- To get into the software, see https://github.com/openSUSE/kiwi
- SUSE Studio provides a development environment for images. It also provides the generated kiwi configuration files for download: http://susestudio.com.

At the time of writing, all the KIWI images had a hardcoded bootloader timeout of 200 seconds. The actual `boottimeout` parameter is put into place after the first boot has happened. It is likely that in the meantime, there is an update for KIWI available that removes that limitation.

Part 4

Other topics

This part of the book includes the following chapters:

21

z/VM live guest relocation

"*If the facts don't fit the theory, change the facts.*"

— Albert Einstein

z/VM 6.2 and later have the ability to relocate Linux guests between members in a single system image (SSI) cluster. This is known as *live guest relocation* (LGR). While continuing to run, Linux systems can be moved cross-LPAR on the same CEC, or cross-CEC, if the SSI is set up that way. This new function allows for very few or even no planned outages.

In this chapter, we provide a brief overview of LGR and information about how to relocate a Linux guest.

21.1 LGR considerations

An SSI cluster has two types of virtual machines:

- Single-configuration virtual machine

  A virtual machine defined by the `USER` statement can be logged on to any member of the SSI cluster, but on only one member at a time. Single-configuration virtual machines are eligible for guest relocation.

- Multi-configuration virtual machine

  A virtual machine defined by the `IDENTITY` and `SUBCONFIG` statements can be logged on concurrently to multiple members of the SSI cluster. The virtual machines have common attributes but can also be configured to access different resources. *Multi-configuration virtual machines are not eligible for guest relocation*.

There are many items that must be considered with regard to relocating running Linux systems:

21.1.1 General considerations before relocation

When determining the size of a guest being relocated, you must take into consideration the following factors:

- The private virtual disks that the virtual machine can have.
- The potential size to which the guest could grow, including standby and reserved memory (storage) settings.
- The level of memory over-commitment currently on the destination system. Relocation may increase paging demands. Therefore, be sure that there is at least two times more paging space than the total virtual memory across all guests.
- A guideline is to never allow paging space for z/VM to go above 50% full. This gives CP space to react to sudden increases in central memory demand. Check on this value with the **`CP QUERY ALLOC PAGE`** command. If you add in the size of the virtual machine being relocated to the pages in use, and that brings the “in use” percentage above 50%, the relocation may have a negative impact on system performance.
- Use the **`VMRELOCATE TEST`** command before **`VMRELOCATE MOVE`**.
- The **`SET RESERVED`** setting for the guest (if any) on the source system is not carried over to the destination system. This setting for the guest on the destination should be established after the relocation completes, which is based on the available resources and workload on the destination system.

21.1.2 Mandatory memory checking performed during relocation

As part of eligibility checking and in-between memory move passes, relocation ensures that the current memory size of Linux fits in available space on the destination system:

- For purposes of the calculation, relocation assumes that the Linux memory is fully populated (including the guest's private virtual disks), and includes an estimate of the size of the supporting CP structures.
- Available space includes the sum of available central, expanded, and auxiliary memory.

There is no way for this check to be bypassed. If it fails, the relocation is terminated. The error message displayed indicates the size of the guest along with the available capacity on the destination system.

21.1.3 Optional memory checking performed during relocation

In addition to the mandatory test described above, by default, the following three checks are also performed during eligibility checking and in-between memory passes:

- Will the guest's current memory size (including CP supporting structures) exceed auxiliary paging capacity on the destination?
- Will the guest's maximum memory size (including CP supporting structures) exceed available space (main storage, expanded storage, and auxiliary storage) on the destination?
- Will the guest's maximum memory size (including CP supporting structures) exceed auxiliary paging capacity on the destination?

> **Note:** The maximum memory size includes any standby and reserved memory that the guest might have.

If any of these tests fail, the relocation is terminated. The error message that is displayed indicates the size of the guest along with the available capacity on the destination system.

If you are certain the above three checks are not-applicable to your installation (for instance, because you have an over-abundance of central memory and a less than recommended amount of paging space), you can choose to have CP skip these three checks by specifying **`FORCE STORAGE`** on the **`VMRELOCATE`** command.

21.1.4 Minimizing link and resource contention

The relocation process monitors system resources and might determine a relocation needs to be slowed down temporarily to avoid exhausting system resources. Link and resource contention may negatively affect performance and thus increase quiesce time during relocation. Therefore, it is recommended only one relocation be performed at a time. If a set of relocations is to be initiated from a single script or EXEC, this can be accomplished by using the **`SYNC`** option (the default) on the **`VMRELOCATE`** command.

21.2 Relocate a Linux system

You can use the **`VMRELOCATE`** command to move a Linux system from the SSI member on which it is running to another member in the cluster. To accomplish this task, perform the following steps:

- Log on as `MAINT` on the member where the Linux system is running. In this example, the Linux system `LINUX1` is running on member 1, `ZVM63A`.
- Choose a sample Linux system to relocate and verify that it is running on the member. In this example, the target is `LINUX1`:

  ```
  ==> q LINUX1
  LINUX1 - DSC
  ```

  The output shows `LINUX1` as disconnected, which means it is running on *this member*.

- Issue the **VMRELOCATE TEST** command with a target of the second SSI member. This will test to see if the system is eligible for relocation:

```
==> vmrelo test linux1 zvm63b
User LINUX1 is eligible for relocation to ZVM63B
Ready; T=0.01/0.01 10:52:06
```

- You may choose to start a **ping** from another session. For example, to ping continuously from a DOS session, issue the following command:

```
c:\>ping /t virtcook1.itso.ibm.com

Pinging virtcook1.itso.ibm.com [9.12.7.1] with 32 bytes of data:

Reply from 9.12.7.1: bytes=32 time=4ms TTL=64
Reply from 9.12.7.1: bytes=32 time=3ms TTL=64
Reply from 9.12.7.1: bytes=32 time=3ms TTL=64
...
```

- Issue the **VMRELOCATE MOVE** command to migrate the running Linux system:

```
==> vmrelo move linux1 zvm63b
Relocation of LINUX1 from ZVM63A to ZVM63B started
User LINUX1 has been relocated from ZVM63A to ZVM63B
```

- Monitor the **ping** session to see if packets are delayed or dropped.
- Verify that the Linux system is now running somewhere in the SSI:

```
==> q LINUX1
LINUX1 - SSI
```

  The output shows `LINUX1` as `SSI`, which means it is running on *a different member*.

This section has shown how to migrate a running Linux system using the **VMRELOCATE** command.

22

DirMaint, SMAPI, and RACF

"*Science is a wonderful thing if one does not have to earn one's living at it.*"

— Albert Einstein

This chapter describes how to enable and configure DirMaint, a directory maintenance product; the z/VM Systems Management APIs (SMAPI); and RACF, a z/VM External Security Manager (ESM).

If you turn on DirMaint, you can no longer edit the `USER DIRECT` file and use the **`DIRECTXA`** command, rather, the DirMaint interface is used.

If you want to turn on SMAPI, which is required by some systems management solutions, you must also have a Directory Maintenance product configured. DirMaint is described here, but also CA products, such as *VM:Secure*, are popular.

Some organizations' security policy require an *External Security Manager* (ESM). RACF is described here, and CA's VM:Secure is also an ESM.

This chapter consists of the following sections:

22.1 Enable and configure DirMaint

To set up DirMaint, perform the following tasks:

1. "Enable DirMaint" on page 372
2. "Configure DirMaint" on page 373
3. "Customize the EXTENT CONTROL file" on page 375
4. "Start DirMaint" on page 377
5. "Test DirMaint" on page 378
6. "Test DirMaint at IPL time" on page 380

22.1.1 Enable DirMaint

In order to use DirMaint, you need a valid license for it. Verify that you are licensed before proceeding.

To install and configure DirMaint, perform the following steps:

1. Log on to MAINT630 on member 1 of the single system image (SSI) cluster.
2. Enable DirMaint via the following **SERVICE** command. Verify that the message VMFSRV2760I is displayed:

```
==> service dirm enable
...                                                     // a few screens go by
VMFSRV1233I The following products have been serviced.
VMFSRV1233I DIRM
VMFSRV2760I SERVICE processing completed successfully
```

3. Put DirMaint into production with the **PUT2PROD** command. Verify that the message VMFP2P2760I is displayed:

```
==> put2prod dirm
VMFP2P2760I PUT2PROD processing started
VMFP2P2760I PUT2PROD processing started for DIRM
VMFP2P1233I The following products have been put into production. Recycle the
            appropriate servers.
VMFP2P1233I DIRM
VMFP2P2760I PUT2PROD processing completed successfully
```

4. This process appends to the end of your SYSTEM CONFIG file. To see the changes, link to the PMAINT CF0 disk and type the SYSTEM CONFIG file to observe these lines at the end of the file:

```
==> vmlink pmaint cf0
DMSVML2060I PMAINT CF0 linked as 0120 file mode Z
==> type system config z
...    // many screens cleared
PRODUCT PRODID 6VMDIR30 STATE ENABLED DESCRIPTION '06/14/12.10:57:20.MAINT630
In
stall/service DirMaint using minidisk'
```

5. Log off from MAINT630.
6. **Repeat** the **PUT2PROD DIRM** command from MAINT630 on every other member in the SSI cluster.

DirMaint should now be enabled on the SSI.

22.1.2 Configure DirMaint

To configure DirMaint, perform the following steps:

1. Change the passwords of certain virtual machines so they can be logged on to:
   a. Log on to MAINT on the first member of the SSI cluster.
   b. Change the passwords of DIRMAINT, DIRMSAT, DIRMSATx (where x is 2, 3, 4 depending on the number of SSI members), DATAMOVE, and DATAMOVx from AUTOONLY to your chosen password:

   ```
   ==> x user direct
   ====> /user dirmaint
   USER DIRMAINT DIRMAINT  128M  256M BDG
    IPL CMS PARM AUTOCR
   ```

   c. Run the **DIRECTXA** command as MAINT on all members to bring the changes online.
2. Log on to 6VMDIR30 on the first member of the SSI cluster.
3. Access the 492 disk as E to get access to the **DIR2PROD EXEC**:

   ```
   ==> acc 492 e
   ```

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

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

5. You have three new minidisks accessed as J, K, and L:

   ```
   ==> q disk
   LABEL  VDEV M  STAT   CYL TYPE BLKSZ   FILES  BLKS USED-(%) BLKS LEFT  BLK TOTAL
   DRM191 191  A   R/W     9 3390 4096        2         12-01       1608       1620
   MNT5E5 5E5  B   R/O    18 3390 4096      133       1666-51       1574       3240
   MNT51D 51D  D   R/W    26 3390 4096      270       1474-31       3206       4680
   DIR1DF 1DF  J   R/W    12 3390 4096       13         20-01       2140       2160
   DRM492 492  K   R/W    15 3390 4096      291       1717-64        983       2700
   DRM41F 41F  L   R/W    16 3390 4096       54        689-24       2191       2880
   MNT190 190  S   R/O   207 3390 4096      698      22295-60      14965      37260
   MNT19E 19E  Y/S R/O   500 3390 4096     1126      29766-33      60234      90000
   ```

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

   ```
   ==> vmlink maint 2cc
   ENTER READ PASSWORD:
   DMSVML2060I MAINT 2CC linked as 0120 file mode Z
   ```

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

   ```
   ==> copy user direct z = input j
   ```

8. Create the main DirMaint configuration file, CONFIGAA DATADVH L. The L disk should be DIRMAINT 41F, which is the pre-production disk. Add the following lines:

   ```
   ==> x configaa datadvh l
   ====> a 10
   ALLOW_ASUSER_NOPASS_FROM= VSMGUARD *
   ALLOW_ASUSER_NOPASS_FROM= VSMWORK1 *
   ALLOW_ASUSER_NOPASS_FROM= VSMWORK2 *
   ```

```
ALLOW_ASUSER_NOPASS_FROM= VSMWORK3 *
ASYNCHRONOUS_UPDATE_NOTIFICATION_EXIT.TCP= DVHXNE EXEC
ASYNCHRONOUS_UPDATE_NOTIFICATION_EXIT.UDP= DVHXNE EXEC
DISK_CLEANUP= YES
ONLINE= IMMED
RUNMODE= OPERATIONAL
RACF_RDEFINE_VMBATCH_DEFAULTS=
```

Notes:

- The ALLOW_ASUSER_NOPASS_FROM lines allow SMAPI users to issue commands to the Directory Manager by using the ASUSER modifier and the password of that user.
- The ASYNCHRONOUS_UPDATE_NOTIFICATION_EXIT lines activate an exit that notifies SMAPI of changes that are made to the user directory.
- If privacy of residual data is a concern on your system, use DISK_CLEANUP= YES.
- The ONLINE= IMMED line sets your changes to be made immediately.
- The RUNMODE= OPERATIONAL line sets directory changes to be made. This can be set to **TESTING** and the changes will not be made.

9. The RACF_RDEFINE_VMBATCH_DEFAULTS= line will not create a VMBATCH-specific resource entry. Otherwise, DIRMAINT will create a VMBATCH resource for this user ID with this line as a default. The VMBATCH generic resource class is configured in the RACF section that is following. If you are not installing RACF this line can be omitted.
10. Create the AUTHFOR CONTROL file on the J disk (DIRMAINT 1DF). Add 12 lines so that MAINT and LNXADMIN are authorized to perform DirMaint tasks, as well as the SMAPI virtual machines VSMGUARD, VSMWORK1, VSMWORK2, and VSMWORK3.

```
==> x authfor control j
====> a 12
ALL LNXADMIN * 140A ADGHOPS
ALL LNXADMIN * 150A ADGHOPS
ALL MAINT * 140A ADGHOPS
ALL MAINT * 150A ADGHOPS
ALL VSMGUARD * 140A ADGHOPS
ALL VSMGUARD * 150A ADGHOPS
ALL VSMWORK1 * 140A ADGHOPS
ALL VSMWORK1 * 150A ADGHOPS
ALL VSMWORK2 * 140A ADGHOPS
ALL VSMWORK2 * 150A ADGHOPS
ALL VSMWORK3 * 140A ADGHOPS
ALL VSMWORK3 * 150A ADGHOPS
```

A command level of 140A allows the authorized user to enter commands using DirMaint Release 4 compatibility syntax. A command level of 150A allows the authorized user to enter commands using the DirMaint Release 5 full function syntax. It is recommended to give access to include records for both 140A and 150A command levels for each target ID/authorized user pair.

You have now created many of the DirMaint configuration files. The next important file is the EXTENT CONTROL file.

22.1.3 Customize the EXTENT CONTROL file

The EXTENT CONTROL file defines disks (volumes) to DirMaint for minidisk allocation. It also contains system and device default values used during allocation operations. There are two main sections that should be populated:

Regions Defines disks and their sizes to DirMaint. The AUTOR keyword can be used in user directory entries to take space from the regions.

Groups Defines pools of disks so the AUTOG keyword can be used to take space from the pools, not from specific disks.

To configure the EXTENT CONTROL file, perform the following steps:

1. Make a copy of the original file:

```
==> copy extent control j = contorig =
```

2. From a different 3270 emulator session to MAINT, you might want to use the **QUERY DASD** command to see which disks are attached to SYSTEM. Disregard the CP-owned DASD and the common volumes:

```
==> q da
DASD 1030 CP OWNED  JV1030   108
DASD 1031 CP OWNED  JS1031   1
DASD 1032 CP OWNED  JP1032   0
DASD 1033 CP SYSTEM JV1033   23
DASD 1034 CP SYSTEM JV1034   1
DASD 1035 CP SYSTEM JV1035   1
DASD 1036 CP OWNED  JV1036   25
DASD 1037 CP SYSTEM JV1037   7
DASD 1038 CP OWNED  JP1038   0
DASD 1039 CP SYSTEM JV1039   0
DASD 103A CP OWNED  JP103A   0
DASD 1131 CP OWNED  JS1131   0
DASD 1136 CP SYSTEM JV1136   16
DASD 1137 CP SYSTEM JV1137   8
DASD 113A CP SYSTEM JM113A   0
DASD 1260 CP SYSTEM JM1260   0
DASD 1261 CP SYSTEM JM1261   0
DASD 1262 CP SYSTEM JM1262   3
DASD 1263 CP SYSTEM JM1263   1
DASD 1264 CP SYSTEM JM1264   4
DASD 1265 CP SYSTEM JM1265   0
DASD 1266 CP SYSTEM JM1266   2
DASD 1267 CP SYSTEM JM1267   2
DASD 1268 CP SYSTEM JM1268   2
DASD 1269 CP SYSTEM JM1269   0
DASD 126A CP SYSTEM JM126A   0
DASD 126B CP SYSTEM JM126B   0
DASD 1360 CP SYSTEM JM1360   0
DASD 1361 CP SYSTEM JM1361   0
DASD 1362 CP SYSTEM JM1362   0
DASD 1363 CP SYSTEM JM1363   0
DASD 1364 CP SYSTEM JM1364   0
DASD 1365 CP SYSTEM JM1365   0
DASD 1366 CP SYSTEM JM1366   0
DASD 1367 CP SYSTEM JM1367   0
DASD 1369 CP SYSTEM JM1369   0
```

```
DASD 136A CP SYSTEM JM136A   0
DASD 136B CP SYSTEM JM136B   0
DASD 136F CP SYSTEM JM136F   0
```

3. Add the DASD attached to `SYSTEM` to the `:REGIONS.` section (assuming these volumes will be available for minidisk creation). The convention used in this example is that the `RegionID`, field 1, is set to the `VolSer`, field 2. Fields 3 and 4 set the cylinder range to all cylinders except cylinder 0, and the `Dev-Type`, the last field, informs DirMaint as to the size of the disk. If you are not sure of the device type, use the **`QUERY DASD DETAILS <rdev>`** command from `MAINT`. Each region name is also added to a group named `POOL1`:

```
==> x extent control j
* *********************************************************************
...

 Purpose:   Default Extent Control file.
...
* *********************************************************************
:REGIONS.
*RegionId  VolSer    RegStart      RegEnd  Dev-Type  Comments
JM113A     JM113A    0001          END     3390-03
JM1262     JM1262    0001          END     3390-09
JM1263     JM1263    0001          END     3390-09
JM1264     JM1264    0001          END     3390-09
JM1265     JM1265    0001          END     3390-09
JM1266     JM1266    0001          END     3390-09
JM1267     JM1267    0001          END     3390-09
JM1268     JM1268    0001          END     3390-09
JM1269     JM1269    0001          END     3390-09
JM126A     JM126A    0001          END     3390-09
JM126B     JM126B    0001          END     3390-09
JM1362     JM1362    0001          END     3390-09
JM1363     JM1363    0001          END     3390-09
JM1364     JM1364    0001          END     3390-09
JM1365     JM1365    0001          END     3390-09
JM1366     JM1366    0001          END     3390-09
JM1367     JM1367    0001          END     3390-09
JM1368     JM1368    0001          END     3390-09
JM1369     JM1369    0001          END     3390-09
JM136A     JM136A    0001          END     3390-09
JM136B     JM136B    0001          END     3390-09

:END.
:GROUPS.
*GroupName RegionList
* POOL1 is for Linux virtual machines
POOL1      JM1262 JM1263 JM1264 JM1265 JM1266 JM1267 JM1268 JM1269
POOL1      JM126A JM126B
POOL1      JM1362 JM1363        JM1365 JM1366 JM1367 JM1368 JM1369
POOL1      JM136A JM136B
* POOL2 is for kiwi
POOL2      JM113A JM1364

:END.
:SSI_VOLUMES.
* Added during Installation, Do not remove.
```

```
*VolumeFamily     Member   VolSer
IBM_RES           ZVM63A JV6280
IBM_WORK1         ZVM63A JV6283
IBM_RES           ZVM63B WV639B
IBM_WORK1         ZVM63B WV639E
:END.
:DEFAULT_GROUPS.
*GroupName Member
:END.
:EXCLUDE.
* entry_name Address
MAINT*       012*
MAINT*       013*
PMAINT       013*
SYSDUMP1     012*
SYSDMP*      012*
:END.
:AUTOBLOCK.
* IBM supplied defaults are contained in the AUTOBLK DATADVH file.
* The following are customer overrides and supplements.
*
*DASDType BlockSize Blocks/Unit Alloc_Unit Architecture
:END.
:DEFAULTS.
* IBM supplied defaults are contained in the DEFAULTS DATADVH file.
* The following are customer overrides and supplements.
*
*DASDType Max-Size
:END.
```

4. Update DirMaint configuration:

```
==> dir2prod update_files 6vmdir30 dirm
DIR2PROD: Matched   CONFIG   SAMPDVH  F  with CONFIG   SDV11501 G2
DIR2PROD: Replacing CONFIG   SAMPDVH  F  with CONFIG   SDV11501 G2
DIR2PROD: Matched   CONFIG   DATADVH  F  with CONFIG   SDV11501 G2
...
DIR2PROD: Matched   LINDFLT  DIRECT   J  with LINDFLT  SAMPDVH  H2
DIR2PROD: Leaving   LINDFLT  DIRECT   J  unchanged.
DIR2PROD: Normal Termination.
```

5. Copy CONFIGAA DATADVH to the 11F minidisk:

```
==> access 11f f
==> acc 41f l
==> copy configaa datadvh l = = f
```

6. Log off from 6VMDIR30.

You should now have the EXTENT CONTROL file configured, which is read when DirMaint starts.

22.1.4 Start DirMaint

To start DirMaint, perform the following steps:

1. Log on to MAINT on the first SSI member.
2. Issue the following command, which is really two separate commands. The command on the left half of the “#” (the line-end character) starts DIRMAINT with the **XAUTOLOG** command

and the **SYNC** option that returns control to MAINT. The second command, on the right side of the "#", sets MAINT to be the secondary user of DIRMAINT. This way, DIRMAINT does not have to be logged on to, but MAINT can see its console output:

```
==> xautolog dirmaint sync # set secuser dirmaint *
AUTO LOGON  ***       DIRMAINT USERS = 13
Ready; T=0.01/0.01 12:17:51
HCPCFX6768I SECUSER of DIRMAINT initiated.
Ready; T=0.01/0.01 12:17:51
DIRMAINT: z/VM V6.2.0    2012-06-07 17:58
DIRMAINT:
......................................................................
..
DIRMAINT:
DIRMAINT: PRODUCT:
DIRMAINT: IBM Directory Maintenance Facility for z/VM (DirMaint)
DIRMAINT: 5741-A07 (C) Copyright IBM Corporation 1979, 2011.
DIRMAINT: Function Level 620 Service Level 0000.
DIRMAINT: DMSACC724I 155 replaces A (191)
DIRMAINT: DMSACP723I F (551) R/O
DIRMAINT: DMSACC723I X (01DE) R/W - OS
DIRMAINT:
 DVHPRO2008I ROLE = DIRMAINT
DIRMAINT: HCPMFS057I OPERATOR not receiving; disconnected
DIRMAINT: DVHPRO2008I ROLE = DIRMAINT
DIRMAINT:
 DIRMAINT: DVHPRO2010I TESTING USE OF MSGNOH ...
DIRMAINT: DASD 0192 DETACHED
DIRMAINT: DASD 021F DETACHED
DIRMAINT:
......................................................................
..
DIRMAINT: DIRMAINT POKDEV62. - 2012/06/14; T=0.02/0.02 12:17:51
 DIRMAINT: DVHILZ3510I Starting DVHINITL with directory: USER INPUT E
 DIRMAINT: DVHILZ3510I DVHINITL Parms: BLDMONO NOCRCWARN
...
DIRMAINT: DVHWAI2140I Waiting for work on 12/06/19 at 12:00:57.
```

Watch for errors. Note the message suggesting that the DirMaint directory is being initialized using the file USER INPUT which was copied from USER DIRECT earlier.

3. Turn off the secondary user setting so MAINT will no longer see the DIRMAINT console messages:

```
==> set secuser dirmaint off
DIRMAINT: HCPCFX6769I Your SECUSER terminated by MAINT.
HCPCFX6769I SECUSER of DIRMAINT terminated.
```

DirMaint should now be running and should have read the USER INPUT, CONFIGAA DATADVH, AUTHFOR CONTROL, and EXTENT CONTROL configuration files.

22.1.5 Test DirMaint

To test DirMaint, perform the following steps:

1. Send the **needpass no** command and type in the MAINT password so that a password is not needed for every DirMaint command. Watch for a 0 return code:

```
==> dirm needpass no
```

```
DVHXMT1181R Enter the current logon password of MAINT at ZVM63A for
DVHXMT1181R authentication.  It will not be displayed on the
DVHXMT1181R terminal.  To exit without processing the command, just
DVHXMT1181R press ENTER.

DVHXMT1191I Your NEEDPASS request has been sent for processing to
DVHXMT1191I DIRMAINT at ZVM63A.
Ready; T=0.01/0.01 10:19:09
 DVHREQ2288I Your USEROPTN request for MAINT at * has been accepted.
 DVHBIU3450I The source for directory entry MAINT has been updated.
 DVHBIU3456I Object directory update is not required for this source
 DVHBIU3456I update.
 DVHREQ2289I Your USEROPTN request for MAINT at * has completed; with
 DVHREQ2289I RC = 0.
```

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

2. Issue **DIRMAINT REVIEW** command. You should no longer need to supply the MAINT password. This will send a file to MAINT's reader containing an overview of the directory entry:

```
==> dirm rev
DVHXMT1191I Your REVIEW request has been sent for processing to DIRMAINT
DVHXMT1191I at ZVM63A.
Ready; T=0.01/0.01 10:20:03
 DVHREQ2288I Your REVIEW request for MAINT at * has been accepted.
RDR FILE 0074 SENT FROM DIRMAINT PUN WAS 0029 RECS 0449 CPY  001 A NOHOLD
NOKEEP
 DVHREQ2289I Your REVIEW request for MAINT at * has completed; with RC
 DVHREQ2289I = 0.
```

3. The file number of the file sent to the reader can be used, which in this example is ***12***. Use the **PEEK** command to view the file. The **for *** parameter specifies to view all records:

```
==> peek 74 (for *
IDENTITY MAINT XXXXXXXX 256M 1000M ABCDEFG
DVHRXV3366I The following configurations will be used on SSI nodes.
DVHRXV3366I The following configuration MAINT-1 will be used on SSI node
DVHRXV3366I ZVM63A.
SUBCONFIG MAINT-1
...
```

4. The disk group named POOL1 was created in the EXTENT CONTROL file. To get disk space from that pool, the **AUTOG** keyword can be used. For example, to create a new Linux virtual machine, take the following steps:

```
==> x linux8 direct a

USER LINUX8 LNX4VM   512M 1G G
   INCLUDE LNXDFLT
   MDISK 0100 3390 AUTOG 5008 POOL1  MR READPASS WRITPASS MULTPASS
   MDISK 0101 3390 AUTOG 5008 POOL1  MR READPASS WRITPASS MULTPASS

==> dirm add linux8
PUN FILE 0011 SENT TO   DIRMSAT  RDR AS  0425 RECS 0012 CPY  001 0 NOHOLD
NOKEEP
DVHXMT1191I Your ADD request has been sent for processing to DIRMAINT at
DVHXMT1191I ZVM63B via DIRMSAT.
Ready; T=0.01/0.01 07:06:47
```

```
 DVHREQ2288I Your ADD request for LINUX8 at * has
 DVHREQ2288I been accepted.
...
 DVHREQ2289I RC = 0.
```

This shows that DirMaint is configured and functioning.

22.1.6 Test DirMaint at IPL time

It is recommended that you shut down and re-IPL the system. With z/VM 6.3, there is new code in AUTOLOG1's **PROFILE EXEC** to start DirMaint. To do so, perform the following steps:

1. **If you are sure that you are in a position to do so**, restart the SSI cluster. This can be accomplished from just one member by prefixing the **SHUTDOWN REIPL** command with the supplied SSICMD EXEC:

   ```
   ==> ssicmd shutdown reipl
   ...
   ```

   You will lose your 3270 emulator sessions. You can watch SSI members status on the HMC.
2. After the system comes back, log on **as MAINT**.
3. Run the **QUERY NAMES** command on all SSI members with the **SSICMD** command:

   ```
   ==> ssicmd q n
   ZVM63A:
   DIRMSAT2 - SSI
   FTPSERVE - DSC , LNXADMIN - DSC , TCPIP    - DSC , DIRMAINT - DSC
   DTCVSW2  - DSC , DTCVSW1  - DSC , VMSERVP  - DSC , VMSERVR  - DSC
   VMSERVU  - DSC , VMSERVS  - DSC , OPERSYMP - DSC , DISKACNT - DSC
   EREP     - DSC , OPERATOR - DSC , MAINT    -L0004
   VSM     - TCPIP

   ZVM63B:
   VMSERVP  - SSI , DIRMAINT - SSI
   FTPSERVE - DSC , LNXADMIN - DSC , TCPIP    - DSC , DIRMSAT2 - DSC
   DTCVSW2  - DSC , DTCVSW1  - DSC , VMSERVR  - DSC , VMSERVU  - DSC
   VMSERVS  - DSC , OPERSYMP - DSC , DISKACNT - DSC , EREP     - DSC
   OPERATOR - DSC
   VSM     - TCPIP
   ```

The DIRMAINT and DIRMSAT2 virtual machines are the primary and satellite DirMaint workers.

This shows that DirMaint is started on both SSI members after a z/VM IPL.

22.2 Configure SMAPI

After DirMaint (or another directory maintenance product) is configured, SMAPI can be enabled and configured. To set up SMAPI, perform the following tasks:

1. "Set up basic SMAPI configuration"
2. "Disable support for ensembles" on page 381
3. "Start SMAPI at IPL time" on page 382
4. "Test SMAPI from CMS" on page 384
5. "Test SMAPI from Linux using smaclient" on page 385

22.2.1 Set up basic SMAPI configuration

The following steps need to be performed on only one SSI member:

1. Log on to MAINT on SSI member 1.
2. Grant authority to the VSMGUARD virtual machine to use certain Shared File System (SFS) directories with the following three **GRANT** commands:

```
==> grant authority vmsys:vsmwork1. to vsmguard (write newwrite
==> grant authority vmsys:vsmwork1.data to vsmguard (write newwrite
==> grant authority * * vmsys:vsmwork1. to vsmguard (read
```

3. Access the shared file system VMSYS:VSMWORK1 as your F disk in read/write mode:

```
==> access vmsys:vsmwork1. f (forcerw
```

4. Edit the file VSMWORK1 AUTHLIST on that disk:

```
==> x vsmwork1 authlist f
```

5. Duplicate the last line by putting a double quotation mark in the prefix area:

   Note: It is important to duplicate the line because lines must be 195 characters wide:

```
00001 DO.NOT.REMOVE
DO.NOT.RE
MOVE
00002 MAINT                                                               ALL
00003 VSMPROXY                                                            ALL
" 004 ZVMLXAPP                                                             ALL
```

6. Press **Enter** and the line will be duplicated. Replace the user ID with LNXADMIN and save the file:

```
00001 DO.NOT.REMOVE
DO.NOT.RE
MOVE
00002 MAINT                                                               ALL
00003 VSMPROXY                                                            ALL
00004 ZVMLXAPP                                                            ALL
00005 LNXADMIN                                                             ALL
```

This change will allow the LNXADMIN virtual machine to invoke SMAPI calls.

22.2.2 Disable support for ensembles

Assuming your system will not be ensemble-managed, virtual machines related to ensembles need to be commented out in a certain configuration file. To turn off ensembles, perform the following steps:

1. As MAINT, access the 193 disk as file mode G:

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

2. Access the shared file system vmsys:vsmwork1.data disk read/write as file mode H:

```
==> acc vmsys:vsmwork1.data h (forcerw
```

3. Copy the DMSSISVR NAMES file from MAINT 193 to the SFS disk:

```
==> copy dmssisvr names g = = h
```

4. Edit the DMSSISVR NAMES file and comment out the last four servers in the file by putting asterisks in the first column of each line:

```
==> x dmssisvr names h
====> /ensembles
...
*** the following machines are only available in ensembles ***
**************************************************************

* Default Management Network Server
*:server.VSMREQIM
*:type.REQUEST
*:protocol.AF_MGMT
*:address.INADDR_ANY
*:port.44446

* Primary Vswitch Controller
*:server.DTCENS1
*:type.VCTRL

* Backup Vswitch Controller
*:server.DTCENS2
*:type.VCTRL

* Management Guest
*:server.ZVMLXAPP
*:type.MG
```

These settings will ensure that virtual machines related to ensembles (especially DTCENS1 and DTCENS2) do not start automatically when SMAPI is started.

22.2.3 Start SMAPI at IPL time

To have SMAPI start at IPL time, add one line to the **PROFILE EXEC** on the AUTOLOG1 191 disk. To accomplish this task, perform the following steps:

1. Link the AUTOLOG1 191 disk read/write and access it as file mode I:

```
==> link autolog1 191 1191 mr
DASD 1192 LINKED R/W;
==> acc 1191 i
```

2. Edit the PROFILE EXEC and add one line to start SMAPI:

```
==> x profile exec i
...
/*******************************************************************/
/* Customer processing can be added here                           */
/*******************************************************************/
"CP XAUTOLOG TCPIP"              /* Start TCPIP                  */
"CP SET MDC STOR 0M 256M"        /* Limit minidisk cache in CSTOR   */
"CP SET MDC XSTORE 0M 0M"        /* Disable minidisk cache in XSTOR  */
"CP SET SIGNAL SHUTDOWN 600"     /* Allow guests 10 min to shut down */
"CP XAUTOLOG LNXADMIN"           /* Start the Linux admin machine     */
"CP XAUTOLOG VSMGUARD"           /* Start SMAPI                     */
...
```

3. **Repeat the previous steps** for all other members in the SSI cluster.

Verify that SMAPI comes up at IPL time

Perform these steps to verify that SMAPI comes up after an IPL:

1. Query the virtual machines running with the **SSICMD EXEC** and the **QUERY NAMES** command to query all active virtual machines on all members:

```
==> ssicmd q n
ZVM63A:
DIRMSAT2 - SSI
FTPSERVE - DSC , LNXADMIN - DSC , TCPIP    - DSC , DIRMAINT - DSC
DTCVSW2  - DSC , DTCVSW1  - DSC , VMSERVP  - DSC , VMSERVR  - DSC
VMSERVU  - DSC , VMSERVS  - DSC , OPERSYMP - DSC , DISKACNT - DSC
EREP     - DSC , OPERATOR - DSC , MAINT    -L0004
VSM     - TCPIP

ZVM63B:
VMSERVP  - SSI , DIRMAINT - SSI
FTPSERVE - DSC , LNXADMIN - DSC , TCPIP    - DSC , DIRMSAT2 - DSC
DTCVSW2  - DSC , DTCVSW1  - DSC , VMSERVR  - DSC , VMSERVU  - DSC
VMSERVS  - DSC , OPERSYMP - DSC , DISKACNT - DSC , EREP     - DSC
OPERATOR - DSC
VSM     - TCPIP
```

2. *If you are sure that you are in a position to do so*, shut down and re-IPL the SSI cluster:

```
==> ssicmd shutdown reipl
SYSTEM SHUTDOWN STARTED
HCPSHU960I System shutdown may be delayed for up to 630 seconds
VMSERVP : DMS5BC3108I Shutdown Signal received. STOP processing started
VMSERVU : DMS5BC3108I Shutdown Signal received. STOP processing started
...
```

3. When the SSI cluster comes back up, log on as `MAINT` to the first SSI member.

4. Query the virtual machines running with the **SSICMD EXEC** as a reference. The SMAPI virtual machines are shown in bold:

```
==> ssicmd q n
ZVM63A:
DIRMSAT2 - SSI
VSMWORK2 - DSC , VSMWORK1 - DSC , FTPSERVE - DSC , VSMGUARD - DSC
LNXADMIN - DSC , TCPIP    - DSC , DIRMAINT - DSC , DTCVSW2  - DSC
DTCVSW1  - DSC , VMSERVP  - DSC , VMSERVR  - DSC , VMSERVU  - DSC
VMSERVS  - DSC , OPERSYMP - DSC , DISKACNT - DSC , EREP     - DSC
OPERATOR - DSC , LOHCOST  - DSC , VSMEVSRV - DSC , VSMPROXY - DSC
VSMREQIU - DSC , VSMREQI6 - DSC , VSMREQIN - DSC , DTCSMAPI - DSC
PERSMAPI - DSC , VSMWORK3 - DSC , MAINT    -L0004
VSM     - TCPIP

ZVM63B:
DIRMAINT - SSI , VMSERVP  - SSI
LOHCOST  - DSC , VSMEVSRV - DSC , VSMPROXY - DSC , VSMREQIU - DSC
VSMREQI6 - DSC , VSMREQIN - DSC , DTCSMAPI - DSC , PERSMAPI - DSC
VSMWORK3 - DSC , VSMWORK2 - DSC , VSMWORK1 - DSC , FTPSERVE - DSC
VSMGUARD - DSC , LNXADMIN - DSC , TCPIP    - DSC , DIRMSAT2 - DSC
DTCVSW2  - DSC , DTCVSW1  - DSC , VMSERVR  - DSC , VMSERVU  - DSC
VMSERVS  - DSC , OPERSYMP - DSC , DISKACNT - DSC , EREP     - DSC
OPERATOR - DSC
VSM     - TCPIP
```

SMAPI should now be running and configured.

22.2.4 Test SMAPI from CMS

To test SMAPI, a REXX EXEC named **CALLSM1** has been included with the files associated with this book. It should have been copied to the MAINT 191 (A) disk in section 5.8.2, “Copy the CPFORMAT EXEC to the members” on page 82. If it was not copied, you will need to do so in order to complete this section.

To test SMAPI, perform the following steps:

1. Log on to MAINT on member 1.
2. Verify that the **CALLSM1 EXEC** has been copied to the MAINT 191 disk:

```
==> listfile callsm1 *
CALLSM1  EXEC     A1
```

3. Link to the TCPMAINT 592 disk:

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

4. Run the **CALLSM1 EXEC**:

```
==> callsm1

                    buffLen = 57
0000 00000035 00000019 496D6167 655F4465 *    5    Image_De *
0016 66696E69 74696F6E 5F517565 72795F44 * finition_Query_D *
0032 4D000000 00000000 00000000 054D4149 * M            MAI *
0048 4E540000 00032A20 00                * NT    *          *

calling send()
receiving requestId,  buffLen = 4
returned from recv() rc,retvalue =0,4
Request id:= 3756453462

receiving length, buffLen = 4
returned from recv() rc,retvalue =0,4
receiving data, buffLen = 2808
returned from recv() rc,retvalue =0,2808

Request id: 3756453462 Return code:0 Reason code:0 possible outdata len:2792

<COMMAND_DEFINE_CPU=>
<COMMAND_SET_CPUAFFINITY=>
<COMMAND_SET_SHARE=>
<COMMAND_SET_VCONFIG=>
<CONSOLE=VDEV=0009 DEVTYPE=3215 CLASS=T>
...
<VMRELOCATE=>
```

This output shows that SMAPI is working from CMS.

22.2.5 Test SMAPI from Linux using smaclient

The script **smaclient** is a powerful, open source bash wrapper around SMAPI. It is available on the web starting at the following site:

http://download.sinenomine.net/smaclient

To test SMAPI using **smaclient**, perform the following steps:

1. Start a root SSH session on the Linux system running on one LNXADMIN.
2. If your Linux system has access to the Internet, you can get the script directly with the **wget** command:

```
# cd /usr/local/sbin
# wget http://download.sinenomine.net/smaclient/smaclient-1.1
--2013-06-13 09:55:22--  http://download.sinenomine.net/smaclient/smaclient-1.1
...
2013-06-13 09:55:22 (3.20 MB/s) - `smaclient-1.1' saved [332722/332722]
# mv smaclient-1.1 smaclient
```

3. If your Linux system does not have access to the Internet, perform the following steps:
   – Download the script from the previous URL to a workstation.
   – Upload the script from the workstation to one of the LNXADMIN systems to the file directory /usr/local/sbin/smaclient.
4. Make the script executable with the **chmod +x** command and verify that it is in the root's path using the **which** command:

```
# chmod +x smaclient
# which smaclient
/usr/local/sbin/smaclient
```

5. Create the file /etc/smaclient.conf so that IUCV is used to communicate to SMAPI:

```
# cd /etc
# vi smaclient.conf
smhost="IUCV"
```

6. Build the **smiucv** binary with the following command. In order to build it, you will need the GNU collection of compilers (gcc) installed:

```
# smaclient smiucv
smiucv built as /usr/local/sbin/smiucv

Make sure "/usr/local/sbin" is included in PATH.
```

   If you do not have gcc, you might first need to run the command **yum install gcc** on RHEL, or **zypper install gcc** on SLES.
7. Test a SMAPI call using **smaclient**. The argument **Image_Query_DM** in the command that follows calls the SMAPI that queries a user directory entry, in this example LNXADMIN:

```
# smaclient Image_Query_DM -T lnxadmin
IDENTITY LNXADMIN LNX4VM 512M 4G BDEG
06130733
   INCLUDE LNXDFLT
06130733
   BUILD ON ZVM63A USING SUBCONFIG LNXADM-1
06130733
   BUILD ON ZVM63B USING SUBCONFIG LNXADM-2
06130733
```

```
   IUCV ANY
06130733
   OPTION MAXCONN 128 LNKNOPAS
06130733
...
```

This shows that SMAPI is running, that `LNXADMIN` is properly authorized to call SMAPI and that the Linux interface **`smaclient`** is working.

22.3 Enable and configure RACF

This section assumes that a new RACF database is being created. For migrating an existing RACF database, see the *RACF Program Directory*, on the web at the following site:

http://www.vm.ibm.com/progdir/6vmrac30.pdf

This section also assumes that DirMaint and SMAPI have been configured according to the previous two sections in this chapter. To configure RACF on a new z/VM 6.3 system, perform the following steps. The first five steps are done before RACF is started. Steps six and seven put RACF into production. The last step is performed after RACF is in production:

1. “Create the RACF command file” on page 387
2. “Customize SMF” on page 389
3. “Delete the ICHRCX02 exit” on page 391
4. “Copy the RACF databases” on page 393
5. “Set up the AUTOLOG1 and AUTOLOG2 virtual machines” on page 398
6. “Enable RACF” on page 398
7. “Put RACF into production on all members” on page 399
8. “Configure SMAPI to work with RACF” on page 405

Important: If you plan to enable RACF, consider Alan Altmark's Words of Wisdom:

1. You must decide on the set of activities you want to audit, and whether audit is always on for those activities or only on demand. It will be necessary to **LINK** and **ACCESS** the active SMF disk to see how fast it is filling. In a Linux farm, most of the activity will be the system programmers and system administrators doing what they do.
2. If both the primary and secondary SMF minidisks unexpectedly become full, then no more audit records can be recorded, even though security-relevant events can continue to occur. Naturally, any such loss of audit records is unacceptable in a secure system. The `SEVER YES` setting in the `SMF CONTROL` file instructs RACF to *sever* when this happens. It is there to ensure "If it didn't get written down, it didn't happen." An excellent policy to have if you are being cross-examined on the witness stand (possibly as the Accused) in a data theft case.
3. The SMF log disks need to be sized to hold an audit log that has all of the data for a single archive interval. That is, if `RACFSMF` is logged on once a day, then the SMF disks need to be large enough to hold one day's worth of data. (Because there are two disks, it can actually hold double that amount per day.)
4. The `RACFSMF 192` archive disk needs to be large enough to hold 'n' archives, where 'n' is your defined value. This is a safety mechanism. The oldest files need to be erased as required to make room for the latest archive. Warning: As shipped, RACFSMF is not this smart. It simply sends a message to `OPERATOR` when the disk is 80% full. How quaint.
5. You must modify RACFSMF to send the newly archived file to a more permanent location. It can FTP it, put it in SFS, SENDFILE to IBM MVS™, dump to tape, FLASHCOPY the 192 to the next in a series of disks,... whatever. It would be useful to have some pre-packaged skeleton activities in SMFPROF.

22.3.1 Create the RACF command file

To set up the initial RACF database, a set of RACF commands is constructed from the user directory source file, then modified later. The **RPIDIRCT EXEC** helps you migrate the user directory data to a RACF database. It translates directory statements into RACF commands and puts them in an output file named `RPIDIRCT SYSUT1`.

To create `RPIDIRCT SYSUT1` for later use with **RPIDIRCT**, perform the following steps:

1. Log on to `MAINT` on the first SSI member.
2. Link the `6VMRAC30 191` disk read/write and access it as file mode `F`:

```
==> link 6vmrac30 191 1191 mr
==> acc 1191 f
```

3. Link the `6VMRAC30 505` disk read/write and access it as file mode `G`:

```
==> link 6vmrac30 505 1505 mr
==> acc 1505 g
```

4. If you are using DirMaint, get the current user directory with passwords with the **DIRMAINT USER WITHPASS** command:

```
==> dirm user withpass
DVHXMT1191I Your USER request has been sent for processing to DIRMAINT
DVHXMT1191I at POKDEV62.
 DVHREQ2288I Your USER request for MAINT at * has been accepted.
```

```
RDR FILE 0004 SENT FROM DIRMAINT PUN WAS 0005 RECS 4539 CPY  001 A NOHOLD
NOKEEP
 DVHREQ2289I Your USER request for MAINT at * has completed; with RC = 0.
```

Receive the file onto the `6VMRAC30 191` disk (F). In this example the reader file was number 4 noted from the previous command output:

```
==> receive 4 = = f
File USER WITHPASS F0 created from USER WITHPASS A0 received from DIRMAINT at
PO
KDEV62
```

5. If you are not using DirMaint, copy USER DIRECT:

```
==> copy USER DIRECT C = = F
```

6. Create the `RPIDIRCT SYSUT1` file from the user directory with the **RPIDIRCT** command. Enter "**n**" to the question of changing the default group ID. This will allow RACF to give all existing virtual machines access to the resources they currently have.

   You might want to issue a **#CP TERM MORE 0 0** because there will be many, many screens of output:

   If you used DirMaint to get the user directory, use:

```
==> rpidirct user withpass f
```

   If you used USER DIRECT file, run:

```
==> rpidirct user direct f

Output defaulted to "A" disk.
  Default group ID = SYS1.
  Would you like to change this default?
  Enter Y/N
n
  Default group ID = SYS1.

PROFILE IBMDFLT

PROFILE TCPCMSU

...
**********  4859 Directory records processed  *********

***************  RPIDIRCT SYSUT1 CREATED  **************
```

7. Make a copy of the newly created `RPIDIRCT SYSUT1` file to have a reference:

```
==> copy rpidirct sysut1 a = sysuorig =
```

8. In the newly created `RPIDIRCT SYSUT1` file, remove all lines with the text `VMBATCH`. A generic `VMBATCH` profile will be created shortly. All lines can be deleted with the **ALL** subcommand and the prefix command **d*** (hidden lines will not be deleted):

```
==> x rpidirct sysut1
====> all /VMBATCH/
====> top
d*=== * * * Top of File * * *
===== --------------------  22  line(s) not displayed --------------------
===== RDEFINE VMBATCH $ALLOC$ OWNER($ALLOC$) UACC(NONE)
...
====> all
```

All lines with VMBATCH should now be deleted.

9. Add the following lines to the bottom of the RPIDIRCT SYSUT1 file:

```
====> bot
====> a 4
setropts generic(vmbatch) gencmd(vmbatch)
rdefine vmbatch ** uacc(none)
permit ** class(vmbatch) id(ftpserve vmnfs dirmsat dirmsat2) acc(control)
setropts classact(vmbatch vmmdisk vmcmd vmlan surrogat)
====> file
```

Notes:

- The first two lines make VMBATCH a generic class.
- The third line permits the FTP, NFS, and DirMaint satellite servers to the VMBATCH class. The number of DIRMSAT* entries should correspond to the number of members in the SSI (for example, if you have a four member SSI, add DIRMSAT3 and DIRMSAT4). Permitting the servers to the VMBATCH class will allow them to use the alternate user ID function.

  More information to protect this function can be found in the Protecting Alternate User IDs section of the z/VM V6R2 RACF Security Server Auditor's Guide on the web at the following site:

  http://publib.boulder.ibm.com/cgi-bin/bookmgr/download/HCSR8C10.pdf
- The fourth line activates the classes VMBATCH, VMMDISK, VMCMD, VMLAN, and SURROGAT.

10. Move the file to the 6VMRAC30 191 disk (F) with the following commands:

```
==> copy rpidirct sysut1 a = = f
==> erase rpidirct sysut1 a
```

The modified RPIDIRCT SYSUT1 file should now be on the 6VMRAC30 191 disk.

22.3.2 Customize SMF

One of the reasons that you run RACF on your z/VM system is to be able to audit who is doing what on the system. In order to do that, the audit records must be managed. This is managed through the RACFSMF virtual machine.

To create a **PROFILE EXEC** for the RACFSMF virtual machine, perform the following steps:

1. Link the RACFSMF 191 disk read/write and access it as file mode H:

```
==> link racfsmf 191 2191 mr
==> acc 2191 h
```

2. Copy the sample profile SMFPROF EXEC to the RACFSMF 191 disk (H) as the file PROFILE EXEC:

```
==> copy smfprof exec g profile = h
```

3. Edit the **PROFILE EXEC** and change the value of Smffreq to **AUTO** and Smfswtch to **NO**:

```
==> x profile exec h
====> /Smfdisk
====> =
...
Smfdisk  = 192
Smfpct   = 80
Smfinfo  = 'OPERATOR'     /* Default message receiver        @VA45455*/
```

```
Smffreq  = 'AUTO'          /* Valid values: DAILY, WEEKLY, MONTHLY,    */
                           /*               AUTO              @VA45455*/
Smfday   = 'MONDAY'        /* Valid values: SATURDAY - FRIDAY  @VA45455*/
Smfswtch = 'NO'            /* Valid values: YES NO             @VA45455*/
...
====> file
```

Note: These changes to the RACFSMF PROFILE EXEC will archive SMF data only when the SMF disk is full. If your site requires archiving regularly, you can use this exec and xautolog the user at each interval.

More information about this topic can be found in the *z/VM RACF Security Server Auditors Guide*, in the chapter: *Processing Audit Records* on z/VM on the web at the following site:

http://publib.boulder.ibm.com/cgi-bin/bookmgr/download/HCSR8C10.pdf

The **PROFILE EXEC** is now configured for the RACFSMF virtual machine.

Modify the SMF CONTROL file

To set SEVER YES in the SMF CONTROL file on the RACFVM 191 disk, perform the following steps:

1. Link to the RACFVM 191 disk read/write and access it as file mode I:

```
==> link racfvm 191 3191 mr
==> acc 3191 i
```

2. Edit the SMF CONTROL file and change SEVER NO to **SEVER YES**:

```
==> x smf control i
====> pre off
* * * Top of File * * *
CURRENT 301 K PRIMARY 301 K SECONDARY 302 K 10000 VMSP CLOSE 001 SEVER YES 0
RAC
====> file
```

Setting this value to **YES** will cause RACF to disconnect from CP if the SMF disks are full.

Note: When RACF is disconnected from CP, users will be unable to log on. To fix the full SMF disk, you will need to logon via OPERATOR using its CP password and IPL CMS. From there, you can copy the SMF records and then clear out the SMF records. Then, restart RACFVM.

3. Copy the modified SMF CONTROL file to the RACFSMF 191 (H) disk:

```
==> copy smf control i = = h
```

4. Link the RACMAINT 191 disk read/write and access it as file mode J:

```
==> link racmaint 191 4191 mr
==> acc 4191 j
```

5. Copy the modified SMF CONTROL file to the RACMAINT 191 disk (J) with the **REPLACE** option:

```
==> copy smf control i = = j (rep
```

6. Log off from MAINT.

The SMF configuration of RACF should now be complete.

22.3.3 Delete the ICHRCX02 exit

Modify the RACF exit named ICHRCX02 to not allow alternate users to access resources that can be accessed by the FTP and NFS servers. To modify the exit, the following high-level steps are required:

1. Access the correct 6VMRAC30 minidisks.
2. Extract the highest level (latest) of the 6VMRAC30 component build list.
3. Copy the extracted build list to a new file with an incremented file number.
4. Comment out the ICHRCX02 object from the build list.
5. Update the VVT table to have a pointer to the new build list.
6. Use the build list to build the local mod RACFLPA module.
7. Place the built 6VMRAC30 module into production.

Perform the following specific steps:

1. Log on as 6VMRAC30 on the first SSI member.
2. Issue the following **VMFSETUP** command (1):

```
==> vmfsetup 6vmrac30 racf
VMFSET2760I VMFSETUP processing started for 6VMRAC30 RACF
VMFUTL2205I Minidisk|Directory Assignments:
            String    Mode  Stat  Vdev  Label/Directory
VMFUTL2205I LOCALSAM  E      R/W  2C2   RAC2C2
VMFUTL2205I APPLY     F      R/W  2A6   RAC2A6
VMFUTL2205I           G      R/W  2A2   RAC2A2
VMFUTL2205I DELTA     H      R/W  2D2   RAC2D2
VMFUTL2205I BUILD0    I      R/W  29E   RAC29E
VMFUTL2205I BUILD6    J      R/W  599   RAC599
VMFUTL2205I BUILD4    K      R/W  505   RAC505
VMFUTL2205I BUILD2    T      R/W  590   RAC590
VMFUTL2205I BUILD8    U      R/W  651   RAC651
VMFUTL2205I BASE      V      R/W  2B2   RAC2B2
VMFUTL2205I --------  A      R/W  191   RAC191
VMFUTL2205I --------  B      R/O  5E5   MNT5E5
VMFUTL2205I --------  D      R/W  51D   MNT51D
VMFUTL2205I --------  S      R/O  190   MNT190
VMFUTL2205I --------  Y/S    R/O  19E   MNT19E
VMFSET2760I VMFSETUP processing completed successfully
```

3. Many RACF disks have been accessed with the **QUERY DISK** command:

```
==> q disk
LABEL  VDEV M   STAT   CYL TYPE BLKSZ   FILES  BLKS USED-(%) BLKS LEFT  BLK TOTAL
RAC191 191  A   R/W     25 3390 4096        6         155-03      4345       4500
MNT5E5 5E5  B   R/O     18 3390 4096      133        1666-51      1574       3240
MNT51D 51D  D   R/W     26 3390 4096      270        1474-31      3206       4680
RAC2C2 2C2  E   R/W      9 3390 4096        0           7-00      1613       1620
RAC2A6 2A6  F   R/W      9 3390 4096        8          15-01      1605       1620
RAC2A2 2A2  G   R/W      9 3390 4096        2           9-01      1611       1620
RAC2D2 2D2  H   R/W     70 3390 4096      146       10257-81      2343      12600
RAC29E 29E  I   R/W     10 3390 4096       60         307-17      1493       1800
RAC599 599  J   R/W     31 3390 4096       44        2506-45      3074       5580
RAC505 505  K   R/W     41 3390 4096      132        5137-70      2243       7380
MNT190 190  S   R/O    207 3390 4096      698       22295-60     14965      37260
RAC590 590  T   R/W     63 3390 4096       24        3631-32      7709      11340
RAC651 651  U   R/W      1 3390 4096        2          35-19       145        180
RAC2B2 2B2  V   R/W     85 3390 4096     2330       12607-82      2693      15300
MNT19E 19E  Y/S R/O    500 3390 4096     1126       29766-33     60234      90000
```

4. Obtain the latest level of the **RPIBLLPA EXEC** with the following **VMFSIM** command: (2)

```
==> vmfsim getlvl 6vmrac30 racf tdata :part rpibllpa exc (history
:PART RPIBLLPA EXC00000 BASE-FILETYPE
```

   The output gives the file name and file type of the currently installed RPIBLLPA EXEC (RPIBLLPA EXC00000). You should see the output **BASE-FILETYPE** in the last field. In VMSES/E terminology, it means that there has been no service to this part by IBM or locally by a system programmer (no entries in the IBM and Local Version Vector Tables).

5. Copy the **RPIBLLPA EXEC** from the 2B2 disk (V as found in the previous **Q DISK** output) to the 2C2 (E) disk with an incremented file type. In this example, the exec becomes EXC00001 if EXEC0000 was last): (3)

```
==> copy rpibllpa exec v = EXCL0001 e
```

6. Edit the newly copied file and comment out the five lines to exclude the ICHRCX02 member: (4)

```
==> x rpibllpa EXCL0001 e
====> /ichrcx02
...
*:OBJNAME. ICHRCX02 LEPARMS RENT REUS LET NCAL XREF DCBS SIZE 100K,80K
*:OPTIONS. CONCAT SYSLIB RACFOBJ
*:PARTID. ICHRCX02 TXT
*:OPTIONS. ENTRY ICHRCX02
*:EOBJNAME.
*
:OBJNAME. ICHSFR00 LEPARMS RENT REUS LET NCAL XREF DCBS SIZE 100K,80K
...
====> file
```

7. Log the local modification to the RPIBLLPA EXEC into the local version vector table with the following **VMFSIM** command: (5)

```
==> vmfsim logmod 6vmrac30 vvtlcl e tdata :mod lcl0001 :part rpibllpa exc
```

8. The 2C2 disk (E) should now contain 6VMRAC30 VVTLCL and RPIBLLPA EXCL0001 files:

```
==> listfile * * e
RPIBLLPA EXCL0001 E2
6VMRAC30 VVTLCL   E1
```

9. Type the contents of the 6VMRAC30 VVTLCL file:

```
==> type 6vmrac30 vvtlcl e

 :PART.RPIBLLPA EXC :MOD.LCL0001
```

10. Generate a new RACFLPA LOADLIB using the **VMFBLD** command: (6)

```
==> vmfbld ppf 6vmrac30 racf rpibllpa (all)
...
VMFLLB2219I Processing object RPIRACEX
VMFBLD1851I (2 of 2) VMFBDLLB completed with return code 0
VMFBLD2180I There are 52 build requirements remaining
VMFBLD2760I VMFBLD processing completed successfully
```

   Be sure that the success message is issued.

11. Link the RACFVM 305 disk read/write and access it as file mode L. If you did not change the passwords, the link password will be **multiple**:

```
==> link racfvm 305 305 mr
```

```
ENTER MULT PASSWORD:
==> acc 305 l
```

12. Use the **VMFCOPY** command to copy the files from the RACFVM 505 disk (K) to the production disk (L): (7)

```
==> vmfcopy racflpa * k = = l (prodid 6vmrac30%racf replace oldd
```

13. Log off from 6VMRAC30.

The RACF exit ICHRCX02 will now be disabled.

22.3.4 Copy the RACF databases

In an SSI, the RACF database must be shared among all members. If you are just installing RACF in a single z/VM LPAR, you can skip this section, which consists of the following subsections:

- “Copy the RACFVM 200 and 300 minidisks”
- “Change RACFVM to shared disks” on page 395
- “Modify the RACMAINT identity” on page 396
- “Define the shared disks in the SYSTEM CONFIG file” on page 397

Copy the RACFVM 200 and 300 minidisks

To copy the RACFVM 200 and 300 minidisks to the volumes that will be shared, perform the following steps:

1. Log on to the first SSI member as MAINT.

> **Important:** If your SSI is on LPARs at the first level, you must use real volumes for the 200 and 300 RACF database, they cannot be minidisks. Use the smallest volumes that you can get because RACF database does not need many cylinders, even mod-3 should be more than enough in most cases. It is not recommended to use volumes with more than 32760 cylinders.

2. Attach the DASD volumes that will be shared:

```
==> q 103B 113B
DASD 103B NW103B  , DASD 113B NW113B
==> att 103B 113B *
0200 0300 ATTACHED TO MAINT
```

3. Change the label with the **CPFMTXA** command so the second character is “R” to signify RACF. It should not be “M” for minidisk or it will be attached to SYSTEM at z/VM IPL time:

```
==> cpfmtxa 103b jr103b label
...
VOLUME SERIAL NUMBER IS NOW = JR103B

ICK00001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
...
==> cpfmtxa 113b jr113b label
...
VOLUME SERIAL NUMBER IS NOW = JR113B

ICK00001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
...
```

4. Link to the RACFVM 200 and RACFVM 300 disks read-only with the **VMLINK** command:

```
==> vmlink racfvm 200
DMSVML2060I RACFVM 200 linked as 0120 file mode Z
==> vmlink racfvm 300
DMSVML2060I RACFVM 300 linked as 0121 file mode X
```

   The virtual device addresses of the linked disks are 120 (for RACFVM 200) and 121 (for RACFVM 300).

5. Copy the RACFVM 200 disk (120) to the 103B volume with the **DDR** command and the following subcommands:

```
==> ddr
z/VM DASD DUMP/RESTORE PROGRAM
ENTER:
====> sysprint cons
ENTER:
====> in 120 3390
ENTER:
====> out 103b 3390
ENTER:
copy 0 to 16
HCPDDR711D VOLID READ IS RACF
DO YOU WISH TO CONTINUE?  RESPOND YES, NO OR REREAD:
yes
ENTER NEXT EXTENT OR NULL LINE
ENTER:

HCPDDR711D VOLID READ IS JR103B
DO YOU WISH TO CONTINUE?  RESPOND YES, NO OR REREAD:
yes
COPYING   RACF
COPYING DATA  06/10/13 AT 18.49.57  GMT FROM RACF   TO JR103B
INPUT CYLINDER EXTENTS      OUTPUT CYLINDER EXTENTS
      START       STOP         START        STOP
          0         16             0          16
END OF COPY Enter
END OF JOB
```

6. Copy the RACFVM 300 disk (121) to the 113B volume with the **DDR** command and the following subcommands:

```
==> ddr
z/VM DASD DUMP/RESTORE PROGRAM
ENTER:
====> sysprint cons
ENTER:
====> in 121 3390
ENTER:
====> out 113B 3390
ENTER:
====> copy 0 to 16
HCPDDR711D VOLID READ IS RACFBK
DO YOU WISH TO CONTINUE?  RESPOND YES, NO OR REREAD:
yes
ENTER NEXT EXTENT OR NULL LINE
ENTER:
```

```
HCPDDR711D VOLID READ IS JR113B
DO YOU WISH TO CONTINUE?  RESPOND YES, NO OR REREAD:
yes
COPYING   RACFBK
COPYING DATA  06/10/13 AT 18.53.36  GMT FROM RACFBK TO JR113B
INPUT CYLINDER EXTENTS      OUTPUT CYLINDER EXTENTS
      START       STOP         START        STOP
          0         16             0          16
END OF COPY
ENTER:
Enter
END OF JOB
```

The contents of the RACF data sets on the RACFVM 200 and 300 minidisks have now been copied to the real devices (at addresses 103B and 113B in this example).

Change RACFVM to shared disks

Now that the 200 and 300 minidisks from one of the SUBCONFIGs of RACFVM have been copied to the DASD volumes that will be shared, these new disks can replace the individual minidisks. To do this, perform the following steps:

1. Get the user directory entry of the RACFVM-1 SUBCONFIG:

   ```
   ==> dirm for racfvm-1 get
   ...
   ```

2. Receive the file from the reader.
3. Comment out the 200 and 300 disks:

   ```
   ==> x racfvm-1 direct
   SUBCONFIG RACFVM-1
    LINK MAINT 0190 0190 RR * CMS system disk
    LINK MAINT 019D 019D RR * help disk
    LINK MAINT 019E 019E RR * Product code disk
    MDISK 191 3390 1568 009 JV1033 MR READ     WRITE    MULTIPLE
   * MDISK 200 3390 1551 017 JV1033 MW READ     WRITE    MULTIPLE
    MDISK 490 3390 1577 070 JV1033 MR READ     WRITE    MULTIPLE
    MDISK 305 3390 1647 136 JV1033 MR READ     WRITE    MULTIPLE
   * MDISK 300 3390 1783 017 JV1033 MW READ     WRITE    MULTIPLE
    MDISK 301 3390 1800 007 JV1033 MR READ     WRITE    MULTIPLE
    MDISK 302 3390 1807 007 JV1033 MR READ     WRITE    MULTIPLE
   ==> file
   ```

4. Replace the RACFVM-1 SUBCONFIG definition:

   ```
   ==> dirm for racfvm-1 rep
   ...
   ```

5. **Repeat the previous steps** for all other members in the SSI cluster. In this example, only the RACFVM-2 SUBCONFIG also had to be modified.
6. Get the user directory entry of the IDENTITY RACFVM:

   ```
   ==> dirm for racfvm get
   ...
   ```

7. Receive the file from the reader.
8. Add the following two MDISK entries for 200 and 300:

   ```
   ==> x racfvm direct
   ```

```
IDENTITY RACFVM   RACFVM     20M   20M ABCDEGH
 BUILD ON LEFT620 USING SUBCONFIG RACFVM-1
 BUILD ON RIGHT620 USING SUBCONFIG RACFVM-2
* BUILD ON @@member3name USING SUBCONFIG RACFVM-3
* BUILD ON @@member4name USING SUBCONFIG RACFVM-4
 IUCV *RPI PRIORITY MSGLIMIT 100
 IUCV ANY PRIORITY MSGLIMIT 50
 IUCV ALLOW MSGLIMIT 255
 ACCOUNT SYSTEMS
 MACH XA
 IPL 490 PARM AUTOCR
 OPTION QUICKDSP MAXCONN 300
 CONSOLE 009 3215 T OPERATOR
 SPOOL 00C 2540 READER *
 SPOOL 00D 2540 PUNCH A
 SPOOL 00E 1403 A
* Add minidisks 200 and 300 for a shared RACF database
 MDISK 200 3390 DEVNO 103B      MWV READ     WRITE    MULTIPLE
 MDISK 300 3390 DEVNO 113B      MWV READ     WRITE    MULTIPLE
...
```

The `DEVNO` operand on the `MDISK` statement specifies a full-pack minidisk, and allows CP to have no dependency on the volume labels of the disks.

9. Replace the `RACFVM SUBCONFIG` definition:

```
==> dirm for racfvm rep
...
 DVHREQ2289I Your REPLACE request for RACFVM at * has completed; with
 DVHREQ2289I RC = 0.
```

Watch for a return code of 0.

The `RACFVM` virtual machine now references the two shared DASD volumes

Modify the RACMAINT identity

The `IDENTITY RACMAINT` has link modes to the `RACFVM 200` and `300` minidisks of `MR`. They must be changed to **`MW`** in order to share the RACF database. To accomplish this, perform the following steps:

1. Get the user directory entry of the `RACMNT-1 SUBCONFIG`:

```
==> dirm for racmnt-1 get
...
```

2. Receive the file from the reader.
3. For the `RACMAINT SUBCONFIG`s, change the link modes to the `RACFVM 200` and `300` disks to from `MR` to **`MW`**. First, is the `RACMNT-1 SUBCONFIG`:

```
==> x racmnt-1 direct
 SUBCONFIG RACMNT-1
  LINK MAINT 0190 0190 RR * CMS system disk
  LINK MAINT 019D 019D RR * help disk
  LINK MAINT 019E 019E RR * Product code disk
  LINK 6VMRAC30 590 490 MR
  LINK 6VMRAC30 505 305 MR
  LINK 6VMRAC30 29E 29E RR
  LINK 6VMRAC30 191 192 RR
  LINK RACFVM 200 200 MW
```

```
  LINK RACFVM 300 300 MW
  LINK RACFVM 301 301 MR
  LINK RACFVM 302 302 MR
==> file
```

4. Replace the user directory entry:

```
==> dirm for racmnt-1 rep
...
```

5. **Repeat the previous steps** for all other members in the SSI cluster. In this example, two member SSI cluster, only the RACMNT-2 SUBCONFIG had to be modified.

The RACF database should now be able to be shared on the volumes at real device addresses 103B and 113B.

Define the shared disks in the SYSTEM CONFIG file

To define the RACF database DASD to CP as devices that can be shared concurrently between real systems, you must add the RDEVICE statements to the SYSTEM CONFIG file.

To do this, perform the following steps:

1. Verify that you are logged on as MAINT.
2. Access the PMAINT CF0 disk read/write. Use the **LINK** command with multi-read (**MR**) parameter:

```
==> link pmaint cf0 cf0 mr
```

3. Use the **ACCESS** command to access it as F:

```
==> acc cf0 f
```

4. Make a copy of the working SYSTEM CONFIG file:

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

5. Edit the original file:

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

6. Add two lines at the bottom specifying that the primary and backup RACF database disks are shared:

```
====> bot
====> a 3
...
/* Define RACF primary and backup databases as shared */
rdevice 103B type dasd shared yes /* RACF primary database */
rdevice 113B type dasd shared yes /* RACF backup database */
```

7. Verify the syntax of the file with your LPAR names as parameter:

```
==> acc 193 g
==> cpsyntax system config f (lpar a02
CONFIGURATION FILE PROCESSING COMPLETE -- NO ERRORS ENCOUNTERED.
==> cpsyntax system config f (lpar a2e
CONFIGURATION FILE PROCESSING COMPLETE -- NO ERRORS ENCOUNTERED.
```

8. Release and detach the PMAINT CF0 (F) disk:

```
==> rel f (det
DASD 0CF0 DETACHED
```

It is also a requirement that CP does not cache data on the RACF database disks in the minidisk cache. Minidisk cache (MDC) is turned off as a result of specifying the DASD as shared in the system configuration file.

The RACF database and backup database should now be shared in the SSI cluster.

22.3.5 Set up the AUTOLOG1 and AUTOLOG2 virtual machines

At z/VM IPL time, the AUTOLOG1 virtual machine normally starts all necessary systems and virtual machines in its **PROFILE EXEC**. When RACF is running, the RACFVM virtual machine must be started first, or other virtual machines will not be able to log in. After the RACF environment is initialized, RACFVM starts the AUTOLOG2 virtual machine, which then starts the remaining servers for the system as AUTOLOG1 normally does. Therefore, the **PROFILE EXEC** needs to be copied from AUTOLOG1 to AUTOLOG2, then modified to start RACFVM.

To accomplish this, perform the following steps:

1. Verify that you are logged on as MAINT on the first member.
2. Link the AUTOLOG1 and AUTOLOG2 191 disks read/write:

```
==> link autolog1 191 1191 mr
==> link autolog2 191 2191 mr
```

3. Access the two disks as file modes F and G:

```
==> acc 1191 f
==> acc 2191 g
```

4. Copy the PROFILE EXEC from AUTOLOG1 to AUTOLOG2:

```
==> copy profile exec f = = g
```

5. Edit the PROFILE EXEC on the AUTOLOG1 191 disk and replace the entire contents with the following to start RACFVM first:

```
==> x profile exec f
/*********************************************************************/
/*  AUTOLOG1 PROFILE EXEC                                            */
/*********************************************************************/
Address Command
"CP XAUTOLOG RACFVM"
"CP LOGOFF"
====> file
```

6. **Perform the steps in the section** on all other SSI members in the cluster.

The AUTOLOG1 virtual machine should now be configured, start RACF (the RACFVM virtual machine). RACF will then start AUTOLOG2 to complete the bootstrapping of the z/VM system.

22.3.6 Enable RACF

To enable RACF, perform the following steps:

1. **Shut down all other members** except the first SSI node. In this example, SSI member 2 was shut down:

```
==> shutdown
...
```

   a. Log on to MAINT630 on the first SSI member.

b. Issue the following **SERVICE** command to enable RACF. This step needs to be performed on only one member. A number of screens will pass by:

```
==> service racf enable
...
VMFSET2760I VMFSETUP processing completed successfully
VMFSRV1233I The following products have been serviced.
VMFSRV1233I CP RACF
VMFSRV2760I SERVICE processing completed successfully
```

RACF should now be enabled on the CF2 disk. This disk is now on the release 1 volume in z/VM 6.3.

2. Shut down the first SSI member:

```
==> shutdown
...
```

RACF should now be enabled and all members and the SSI should be shut down.

22.3.7 Put RACF into production on all members

Important: The next paragraph is extremely important. Read it at least twice.

The **PUT2PROD** command must be run on each member of the SSI. **Start with the first member**. Perform all five of the following subsections on the first member. If you are in an SSI, you will later perform only the **first and last** subsection on the other members:

1. "IPL the member and start RACMAINT"
2. "Configure the initial RACF database"
3. "Enable DirMaint to RACF on the first member"
4. "Set the DirMaint use of the reader with RACF on the first member"
5. "Put RACF into production"

IPL the member and start RACMAINT

You must IPL each member of the SSI and start RACMAINT. To do this, perform the following steps:

1. Start an Integrated 3270 Console for the member.
2. IPL the member from the HMC from the real device address "Res volume".
3. The SAPL window should appear on the Integrated 3270 Console.
4. Change the *Device Number* to that of the **Release Volume 1**. (not the "Res volume" that is normally IPLed). In this example, it was real device address ***1136***. Press F10 to IPL. This will load the **CPLOAD MODULE** from the CF2 disk, which has RACF enabled:

```
STAND ALONE PROGRAM LOADER: z/VM VERSION 6 RELEASE 3.0

 DEVICE NUMBER:   1136     MINIDISK OFFSET:   39         EXTENT:  1

 MODULE NAME:     CPLOAD     LOAD ORIGIN:       1000

 --------------------------------IPL PARAMETERS--------------------------------
fn=SYSTEM ft=CONFIG pdnum=1 pdvol=1036
```

```
----------------------------------COMMENTS----------------------------------



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


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

5. Supply the `NOAUTOLOG` parameter so that the `PROFILE EXEC` on `AUTOLOG1` is not run and `RACFVM` is not started:

```
16:30:25 Start ((Warm|Force|COLD|CLEAN) (DRain) (DIsable)  (NODIRect)
16:30:25       (NOAUTOlog)) or (SHUTDOWN)
noautolog
...
```

6. Continue to IPL the member. When the IPL process completes, you will be logged on as `OPERATOR`. Start the virtual machine `RACMAINT`. You should see messages indicating that the `200` and `300` disks are read/write. If you see errors about them, you have to fix the problem:

```
==> xautolog racmaint
...
```

RACF should now be running on the SSI member with a skeleton database.

If you have already done the next three sections on the first SSI member, proceed to the section “Put RACF into production” on page 404.

Configure the initial RACF database

The following set of steps need to be performed only once to populate and customize the RACF database.

1. **On the first SSI member**, disconnect from `OPERATOR`.

```
==> disc
```

2. Log on to `IBMUSER` with a password of **SYS1**. This is a default virtual machine that is created for RACF configuration.
3. You will see a message that the password has expired. Reset the password by typing in the new password twice separated by a “/”. You will see resource errors; these are expected:

```
LOGON IBMUSER
RPIMGR042I PASSWORD EXPIRED

To change your password - enter: nnn/nnn  where nnn = new password
or,
enter LOGOFF to cancel

 ICH70001I IBMUSER  LAST ACCESS AT **:**:** ON ****, **** **,****
 HCPRPW004I Password changed
RPIMGR031E RESOURCE MAINT.190 SPECIFIED BY LINK COMMAND NOT FOUND
RPIMGR031E RESOURCE MAINT.19E SPECIFIED BY LINK COMMAND NOT FOUND
RPIMGR031E RESOURCE 6VMRAC30.29E SPECIFIED BY LINK COMMAND NOT FOUND
RPIMGR031E RESOURCE 6VMRAC30.505 SPECIFIED BY LINK COMMAND NOT FOUND
RPIMGR031E RESOURCE 6VMRAC30.191 SPECIFIED BY LINK COMMAND NOT FOUND
```

```
RPIMGR031E RESOURCE RACFVM.305 SPECIFIED BY LINK COMMAND NOT FOUND
RPIMGR031E RESOURCE IBMUSER.191 SPECIFIED BY LINK COMMAND NOT FOUND
z/VM Version 6 Release 2.0, Service Level 1101 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES:   NO RDR,   NO PRT,   NO PUN
LOGON AT 13:24:34 EDT FRIDAY 06/22/12
z/VM V6.2.0    2012-06-21 16:54
...
```

4. Set the F12 function key to the command **RETRIEVE**:

```
==> set pf12 ret
```

5. Link and access 6VMRAC30's 505, 191, and 29E disks. Disregard any error messages:

```
==> link 6vmrac30 505 505 rr
RPIMGR031E RESOURCE 6VMRAC30.505 SPECIFIED BY LINK COMMAND NOT FOUND
DASD 0505 LINKED R/O; R/W BY RACMAINT
==> acc 505 c
DMSACP723I C (505) R/O
==> link 6vmrac30 191 192 rr
RPIMGR031E RESOURCE 6VMRAC30.191 SPECIFIED BY LINK COMMAND NOT FOUND
==> acc 192 b
DMSACP723I B (192) R/O
DMSACP725I 192 also = D disk
==> link 6vmrac30 29e 29e rr
RPIMGR031E RESOURCE 6VMRAC30.29E SPECIFIED BY LINK COMMAND NOT FOUND
==> acc 29e d
DMSACP724I 29E replaces D (192) R/O
DMSACP723I D (29E) R/O
```

6. Update the RACF database with existing CP directory information using the **RPIBLDDS** command. The RPIDIRCT SYSUT1 file created earlier and copied to the 6VMRAC30 191 disk is used as input. You may again choose to issue the command **#CP TERM MORE 0 0** as many screens full of messages will be issued:

```
==> rpibldds rpidirct
Processing batch file RPIDIRCT SYSUT1 using “RAC” command interface
...
 => PERMIT LOGONBY.SSLDCSSM CLASS(SURROGAT) ID(TCPMAINT) ACCESS(READ)
 => PERMIT LOGONBY.SSLDCSSM CLASS(SURROGAT) ID(GSKADMIN) ACCESS(READ)
 => setropts generic(vmbatch) gencmd(vmbatch)
 => rdefine vmbatch ** uacc(none)
 => permit ** class(vmbatch) id(ftpserve vmnfs dirmsat dirmsat2) acc(control)
 => setropts classact(vmbatch vmmdisk vmcmd vmlan surrogat)
```

The RACF database should now be populated with the values from the user directory, and other modifications configured previously.

7. Define the security administrator virtual machine. In this example, the default of SYSADMIN is used:

```
==> rac alu sysadmin special
```

8. Log off from IBMUSER.
9. Log on to SYSADMIN. You will be asked to change the password.
10. Grant the following virtual machines OPERATIONS privileges:

```
==> rac alu datamove operations
==> rac alu MAINT630 operations
```

```
==> rac alu bldseg operations
==> rac alu lnxadmin operations
```

These commands give the four virtual machines specified access to all minidisks on the system.

11. Revoke the privileges for the IBMUSER virtual machine as it is no longer needed:

```
==> rac alu ibmuser revoke
```

12. Grant the DIRMAINT virtual machine SPECIAL privileges:

```
==> rac alu dirmaint special
```

13. Grant the MAINT virtual machine SPECIAL and OPERATIONS privileges:

```
==> rac alu maint special operations
```

14. Define the system virtual switches named VSW1 and VSW2 to the VMLAN class:

```
==> rac rdefine vmlan system.vsw1
==> rac rdefine vmlan system.vsw2
```

15. Permit TCPIP to the virtual switch VSW1:

```
==> rac permit system.vsw1 class(vmlan) id(tcpip) access(update)
```

16. Permit Linux machines to the virtual switch VSW1:

```
==> rac permit system.vsw1 class(vmlan) id(lnxadmin) access(update)
==> rac permit system.vsw1 class(vmlan) id(linux1) access(update)
==> rac permit system.vsw1 class(vmlan) id(linux2) access(update)
==> rac permit system.vsw1 class(vmlan) id(linux3) access(update)
==> rac permit system.vsw1 class(vmlan) id(linux4) access(update)
==> rac permit system.vsw1 class(vmlan) id(linux5) access(update)
==> rac permit system.vsw1 class(vmlan) id(linux6) access(update)
```

17. Log off from SYSADMIN.

The initial RACF database should now be configured.

Enable DirMaint to RACF on the first member

Perform the following steps to enable DirMaint to run to RACF:

1. Log on to MAINT. You should be asked to change the password.
2. Link to the 6VMDIR30 2C2 disk read-only, which has a sample CONFIGRC DATADVH file:

```
==> vmlink 6VMDIR30 2c2
DMSVML2060I 6VMDIR30 2C2 linked as 0120 file mode Z
```

3. Copy the sample CONFIGRC file from the Z disk to the A disk as file type DATADVH:

```
==> copy configrc sampdvh z = datadvh a
```

4. Start DirMaint with the **XAUTOLOG DIRMAINT** command:

```
==> xautolog dirmaint
 ICH70001I DIRMAINT LAST ACCESS AT 15:38:05 ON WEDNESDAY, JUNE 20, 2012
Command accepted
Ready; T=0.01/0.01 15:50:02
AUTO LOGON  ***       DIRMAINT USERS = 5
HCPCLS6056I XAUTOLOG information for DIRMAINT: The IPL command is verified by
th
e IPL command processor.
 DVHPRO2008I ROLE = DIRMAINT
```

5. Add the CONFIGRC DATADVH configuration file to DirMaint with the **DIRM FILE** command. You can ignore error messages such as the RPIMGR031E message shown below:

```
==> dirm file configrc datadvh
RPIMGR031E RESOURCE DIRMAINT SPECIFIED BY SPOOL COMMAND NOT FOUND
RPIMGR031E RESOURCE POKDEV62 SPECIFIED BY TAG COMMAND NOT FOUND
PUN FILE 0011 SENT TO   DIRMAINT RDR AS  0004 RECS 0103 CPY  001 0 NOHOLD
NOKEEP
DVHXMT1191I Your FILE request has been sent for processing to DIRMAINT
DVHXMT1191I at POKDEV62.
 DVHREQ2288I Your FILE request for MAINT at * has been accepted.
 DVHRCV3821I File CONFIGRC DATADVH A2 has been received; RC = 0.
 DVHREQ2289I Your FILE request for MAINT at * has completed; with RC = 0.
```

6. Issue the **DIRM RLDDATA** command so the change is activated:

```
==> dirm rldd
DVHXMT1191I Your RLDDATA request has been sent for processing to
DVHXMT1191I DIRMAINT at POKDEV62.
DVHREQ2288I Your RLDDATA request for MAINT at * has been accepted.
 DVHITI6314E No DATAMOVE machines were defined in the config file.
 DVHREQ2289I Your RLDDATA request for MAINT at * has completed; with RC =
 DVHREQ2289I 0.
```

DirMaint should now be initially enabled to RACF.

Set the DirMaint use of the reader with RACF on the first member

Because the VMBATCH definitions were deleted in section 22.3.1, "Create the RACF command file" on page 387, RACF will give errors when DirMaint sends files to the reader. To address this issue, the CP **TRANSFER** and **TAG** commands need not be controlled.

In addition, SMAPI needs to issue commands for other users with the **FOR** command under privilege class C. To address this, the CP **FOR.C** commands need *not* be controlled.

To effect these settings, perform the following steps:

1. Create a RACF profile for the VMXEVENT class named EVENT1:

```
==> rac rdefine vmxevent event1
```

2. Add three members to the VMEVENT class for the **TRANSFER** (privilege class G), the **TAG** commands and for the **FOR** (privilege class C), and set them to no-control:

```
==> rac ralter vmxevent event1 addmem(transfer.g/noctl tag/noctl for.c/noctl)
```

3. Activate the VMXEVENT class:

```
==> rac setropts classact(vmxevent)
```

4. Refresh the VMEVENT class:

```
==> rac setevent refresh event1
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: COUPLE
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: FOR.G
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: LINK
...
```

5. Log off from MAINT.

DirMaint and SMAPI should now be enabled to run with RACF.

Put RACF into production

RACF is now configured to be into production. Put RACF into production with the following steps:

1. If you are OPERATOR, disconnect:

```
==> disc
...
```

2. Log on to MAINT630 on the next member. You will be asked to change the password on the first member. On subsequent members, use the new password.
3. Start the AUTOLOG2 virtual machine with the **XAUTOLOG** command to start the shared file pool server machines:

```
==> xautolog autolog2
 ICH70001I AUTOLOG2 LAST ACCESS AT **:**:** ON ****, **** **,****
Command accepted
AUTO LOGON  ***       AUTOLOG1 USERS = 5
HCPCLS6056I XAUTOLOG information for AUTOLOG1: The IPL command is verified by
th
e IPL command processor.
```

4. Put RACF into production with the **PUT2PROD RACF** command. Watch for the completed successfully message:

```
==> put2prod rac
...
Note to editors: output is in RACF-VMVFEWIscreen.
```

5. Put CP into production with the **PUT2PROD CP** command. Watch for the completed successfully message:

```
==> put2prod cp
...                         // a number of screens pass by
VMFP2P2760I PUT2PROD processing completed successfully
```

   RACF should now be prepared to go into production at the next IPL.

6. Log off from MAINT630.
7. Log on to OPERATOR. You will be asked to change the password on the first member.
8. Log the RACMAINT virtual machine off with the **FORCE** command:

```
==> force racmaint
RACMAINT: CONNECT= 00:37:57 VIRTCPU= 000:03.32 TOTCPU= 000:04.03
RACMAINT: LOGOFF AT 16:11:53 EDT WEDNESDAY 06/20/12 BY OPERATOR
16:11:53 USER DSC   LOGOFF AS  RACMAINT USERS = 22    FORCED BY OPERATOR
16:11:53 HCPRPI036E CP/RACF communication path broken to RACMAINT
```

9. Start the RACFVM virtual machine with the **XAUTOLOG** command and watch for messages indicating that RACF is starting:

```
==> xautolog racfvm
14:42:39 Command accepted
14:42:39 AUTO LOGON  ***       RACFVM   USERS = 23    BY OPERATOR
16:12:00 HCPCLS6056I XAUTOLOG information for RACFVM: The IPL command is
verifie
d by the IPL command processor.
RACFVM  : RACFVM CMS XA Rel 14 11/18/2010
RACFVM  : DMSACP723I B (305) R/O
RACFVM  : RACF is defined to the Z/VM system and the current product status is
E
```

```
NABLED
RACFVM  :
RACFVM  :       RACF
RACFVM  :  Feature for z/VM
RACFVM  :  Version 6.2.0
RACFVM  :
RACFVM  :  Licensed Materials - Property of IBM
RACFVM  :  5741-A07
RACFVM  :  (C) Copyright IBM CORP. 1981, 2010  All Rights Reserved.
RACFVM  :
RACFVM  : DMSACC723I R (0200) R/W - OS
RACFVM  : DMSACC723I Q (0300) R/W - OS
...
16:12:02 HCPRPI035I CP/RACF communication path established to RACFVM
...
```

RACF should now be running on the current member.

10. Shut down the member:

```
==> shutdown
...
00: 13:52:25 HCPWRP961W SYSTEM SHUTDOWN COMPLETE FOR LEFT620  ON 2012-06-22
00: HCPGIR450W CP entered; disabled wait PSW 00020000 00000000 00000000
00000961
```

For SSI members other than the first, perform the steps in only the first and last of the five subsections:

"IPL the member and start RACMAINT" on page 399.

"Put RACF into production" on page 404.

When the **PUT2PROD** sections have been performed on all SSI members, IPL the members one at a time from the default (*RES*) volume. Do not specify the NOAUTOLOG parameter. You should see RACF start on the OPERATOR console.

When the system comes back up, RACF should be running.

22.3.8 Configure SMAPI to work with RACF

Perform the following steps to allow SMAPI to work with RACF:

1. You should be able to access your system through a 3270 emulator.
2. Log on to MAINT on the first SSI member.
3. Allow VSMWORK1 to have CONTROL authority the z/VM minidisk (VMMDISK) that contains the SYSTEM CONFIG file (PMAINT CF0), perform the following command:

```
==> rac permit pmaint.cf0 class(vmmdisk) acc(control) id(vsmwork1)
==> rac permit maint.cf1 class(vmmdisk) acc(control) id(vsmwork1)
```

4. Allow VSMWORK1 to have CONTROL access to the generic class VMBATCH:

```
==> rac permit ** class(vmbatch) id(vsmwork1) access(control)
```

5. Allow SMAPI workers to read the TCPMAINT 198 disk:

```
==> rac permit tcpmaint.198 class(vmmdisk) acc(read) id(vsmguard)
==> rac permit tcpmaint.198 class(vmmdisk) acc(read) id(vsmwork1)
==> rac permit tcpmaint.198 class(vmmdisk) acc(read) id(vsmwork2)
```

```
==> rac permit tcpmaint.198 class(vmmdisk) acc(read) id(vsmwork3)
```

6. Allow LNXADMIN to read certain disks:

```
==> rac permit pmaint.cf0 class(vmmdisk) acc(read) id(lnxadmin)
==> rac permit autolog1.191 class(vmmdisk) acc(read) id(lnxadmin)
==> rac permit tcpmaint.198 class(vmmdisk) acc(read) id(lnxadmin)
```

7. Change default password expiration to your security standard; 186 days in this example:

```
==> rac setropts password(interval(186))
```

Enable RACROUTE

Enable the SMAPI service machines VSMREQI6, VSMREQIN, VSMREQIU, VSMEVSRV, DTCSMAPI, VSMWORK1, VSMWORK2, and VSMWORK3 to use **RACROUTE** services with the following commands:

```
==> RAC SETROPTS CLASSACT(FACILITY)
==> RAC RDEFINE FACILITY ICHCONN UACC(NONE)
ICH10006I RACLISTED PROFILES FOR FACILITY WILL NOT REFLECT THE ADDITION(S)
UNTIL
 A SETROPTS REFRESH IS ISSUED.
==> RAC PERMIT ICHCONN CLASS(FACILITY) ID(VSMREQI6) ACCESS(UPDATE)
...
==> RAC PERMIT ICHCONN CLASS(FACILITY) ID(VSMREQIN) ACCESS(UPDATE)
...
==> RAC PERMIT ICHCONN CLASS(FACILITY) ID(VSMREQIU) ACCESS(UPDATE)
...
==> RAC PERMIT ICHCONN CLASS(FACILITY) ID(VSMEVSRV) ACCESS(UPDATE)
...
==> RAC PERMIT ICHCONN CLASS(FACILITY) ID(DTCSMAPI) ACCESS(UPDATE)
...
==> RAC PERMIT ICHCONN CLASS(FACILITY) ID(VSMWORK1) ACCESS(UPDATE)
...
==> RAC PERMIT ICHCONN CLASS(FACILITY) ID(VSMWORK2) ACCESS(UPDATE)
...
==> RAC PERMIT ICHCONN CLASS(FACILITY) ID(VSMWORK3) ACCESS(UPDATE)
...
==> RAC SETROPTS RACLIST(FACILITY)
```

Exempt SMAPI from certain command checking

Four SMAPI service machines (DTCSMAPI, VSMWORK1, VSMWORK2, and VSMWORK3) should be made exempt from access checking. Even if access checking is not active on your system, make the SMAPI service machines exempt from access checking for the **FOR** (privilege class C), and **LINK** commands:

- Make the DTCSMAPI virtual machine exempt with the following commands:

```
==> RAC SETROPTS CLASSACT(VMXEVENT)
==> RAC RDEFINE VMXEVENT USERSEL.DTCSMAPI
==> RAC RALTER VMXEVENT USERSEL.DTCSMAPI ADDMEM(FOR.C/NOCTL)
==> RAC RALTER VMXEVENT USERSEL.DTCSMAPI ADDMEM(LINK/NOCTL)
==> RAC SETEVENT REFRESH USERSEL.DTCSMAPI
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: COUPLE
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: FOR.G
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: STORE.C
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TAG
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TRANSFER.D
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TRANSFER.G
```

```
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TRSOURCE
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG088
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG0A0
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG0D4
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG0E4
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG280
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: APPCPWVL
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: MDISK
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: RSTDSEG
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: RDEVCTRL
RPISET126I SETEVENT COMPLETED SUCCESSFULLY.
```

- Make the VSMWORK1 virtual machine exempt with the following commands:

```
==> RAC RDEFINE VMXEVENT USERSEL.VSMWORK1
==> RAC RALTER VMXEVENT USERSEL.VSMWORK1 ADDMEM(FOR.C/NOCTL)
==> RAC RALTER VMXEVENT USERSEL.VSMWORK1 ADDMEM(LINK/NOCTL)
==> RAC SETEVENT REFRESH USERSEL.VSMWORK1
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: COUPLE
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: FOR.G
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: STORE.C
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TAG
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TRANSFER.D
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TRANSFER.G
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TRSOURCE
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG088
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG0A0
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG0D4
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG0E4
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG280
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: APPCPWVL
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: MDISK
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: RSTDSEG
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: RDEVCTRL
RPISET126I SETEVENT COMPLETED SUCCESSFULLY.
```

- Make the VSMWORK2 virtual machine exempt with the following commands:

```
==> RAC RDEFINE VMXEVENT USERSEL.VSMWORK2
==> RAC RALTER VMXEVENT USERSEL.VSMWORK2 ADDMEM(FOR.C/NOCTL)
==> RAC RALTER VMXEVENT USERSEL.VSMWORK2 ADDMEM(LINK/NOCTL)
==> RAC SETEVENT REFRESH USERSEL.VSMWORK2
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: COUPLE
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: FOR.G
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: STORE.C
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TAG
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TRANSFER.D
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TRANSFER.G
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TRSOURCE
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG088
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG0A0
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG0D4
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG0E4
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG280
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: APPCPWVL
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: MDISK
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: RSTDSEG
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: RDEVCTRL
```

```
RPISET126I SETEVENT COMPLETED SUCCESSFULLY.
```

- Make the VSMWORK3 virtual machine exempt with the following commands:

```
==> RAC RDEFINE VMXEVENT USERSEL.VSMWORK3
==> RAC RALTER VMXEVENT USERSEL.VSMWORK3 ADDMEM(FOR.C/NOCTL)
==> RAC RALTER VMXEVENT USERSEL.VSMWORK3 ADDMEM(LINK/NOCTL)
==> RAC SETEVENT REFRESH USERSEL.VSMWORK3
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: COUPLE
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: FOR.G
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: STORE.C
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TAG
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TRANSFER.D
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TRANSFER.G
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TRSOURCE
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG088
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG0A0
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG0D4
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG0E4
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG280
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: APPCPWVL
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: MDISK
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: RSTDSEG
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: RDEVCTRL
RPISET126I SETEVENT COMPLETED SUCCESSFULLY.
```

RACF should now allow SMAPI to do its job. It is recommended that you go back and try the sections 22.2.4, “Test SMAPI from CMS” on page 384 and 22.2.5, “Test SMAPI from Linux using smaclient” on page 385.

22.4 Verifying DirMaint and RACF to work together

To add new virtual machines, some DirMaint and some RACF commands are needed: Perform the following steps:

1. Log in as MAINT.
2. Create a sample virtual machine prototype named LNXSAMPL PROTODIR:

```
==> x lnxsampl protodir a
USER LNXSAMPL LNX4VM 256M 2G G
    INCLUDE LNXDFLT
    MDISK 0100 3390 AUTOG 10016 POOL1 MR LNX4VM LNX4VM LNX4VM
    MDISK 0101 3390 AUTOG 10016 POOL1 MR LNX4VM LNX4VM LNX4VM
```

   This definition will give each Linux virtual machine 256 MB of initial memory (with up to 2GB dynamic) and 2 3390-9 disks or about 14 GB of disk space. The **AUTOG** and **POOL1** keywords instruct DirMaint to automatically choose space from the pool of volumes in the pool named POOL1.

3. Register the prototype with DirMaint using the **DIRM FILE** command:

```
==> dirm file lnxsampl protodir
10:08:53 PUN FILE 0069 SENT TO   DIRMAINT RDR AS  0086 RECS 0012 CPY  001 0
NOHO
LD NOKEEP
DVHXMT1191I Your FILE request has been sent for processing to DIRMAINT
DVHXMT1191I at POKDEV62.
```

```
 DVHREQ2288I Your FILE request for MAINT at * has been accepted.
 DVHRCV3821I File LNXSAMPL PROTODIR A has been received; RC = 0.
 DVHREQ2289I Your FILE request for MAINT at * has completed; with RC = 0.
```

4. Create a new virtual machine with the **DIRM ADD** command and the **LIKE** parameter. In this example, the user ID is named LINUX8:

```
==> dirm add linux8 like lnxsampl pw lnx4vm
DVHXMT1191I Your ADD request has been sent for processing to DIRMAINT at
DVHXMT1191I POKDEV62.

 DVHREQ2288I Your ADD request for LINUX76 at * has been accepted.
 ...
 DVHSHN3430I AMDISK operation for LINUX76 address 0101 has finished (WUCF
 DVHSHN3430I 07101436).
 DVHREQ2289I Your ADD request for LINUX76 at * has completed; with RC =
 DVHREQ2289I 0.
```

5. Allow the new user access to the virtual switches named VSW1 and VSW2:

```
==> rac permit system.vsw1 class(vmlan) id(linux8) access(update)
==> rac permit system.vsw2 class(vmlan) id(linux8) access(update)
```

This shows an example of DirMaint working with RACF when creating new virtual machines.

22.5 Some common DirMaint tasks

The following common DirMaint tasks are described in the sections that follow:

- "Update a user directory entry"
- "Edit the EXTENT CONTROL file" on page 410
- "Get a copy of the user directory" on page 412
- "Add an IDENTITY" on page 412
- "Clean up work units" on page 413

22.5.1 Update a user directory entry

There are many ways to modify virtual machine or PROFILE directory entries with DirMaint. Perhaps one of the simplest is with GET and REPLACE. The **GET** operand of the **DIRMAINT** command retrieves a copy of a user or profile directory entry. You can then introduce new control statements or modify existing ones. The updated directory entry can be restored to the directory using the **REPLACE** operand. Following is an example of modifying the directory entry for the virtual machine LINUX153:

1. Log on to MAINT.
2. Use the **GET** operand to unlock the directory entry:

```
==> dirm for linux153 get
DVHXMT1191I Your GET request has been sent for processing to DIRMAINT at
DVHXMT1191I POKDEV62.
Ready; T=0.01/0.01 10:14:45
 DVHREQ2288I Your GET request for LINUX153 at * has been accepted.
 DVHGET3304I Directory entry  LINUX153 is now locked.
 RDR FILE 0005 SENT FROM DIRMAINT PUN WAS 0029 RECS 0010 CPY  001 A NOHOLD
NOKEEP
 DVHREQ2289I Your GET request for LINUX153 at * has completed; with RC
 DVHREQ2289I = 0.
```

Always watch for a 0 return code. Note that the file sent is number **5**.

3. Receive the directory entry from the reader to MAINT's A disk with the **REPLACE** option:

```
==> receive 5 (rep
File LINUX153 DIRECT A0 replaced by LINUX153 DIRECT A0 received from DIRMAINT
at
 POKDEV62
```

4. Edit the directory entry and make the wanted changes:

```
==> x linux153 direct
...
```

5. Use the **REPLACE** operand to make the changes effective:

```
==> dirm for linux153 rep
PUN FILE 0006 SENT TO   DIRMAINT RDR AS  0033 RECS 0013 CPY  001 0 NOHOLD
NOKEEP
DVHXMT1191I Your REPLACE request has been sent for processing to
DVHXMT1191I DIRMAINT at POKDEV62.
Ready; T=0.01/0.01 10:14:51
 DVHREQ2288I Your REPLACE request for LINUX153 at * has been accepted.
 ...
 DVHBIU3428I online.
 DVHREP3603I Directory entry LINUX153 is now unlocked.
 DVHREQ2289I Your REPLACE request for LINUX153 at * has completed; with
 DVHREQ2289I RC = 0.
```

This has shown how to modify a virtual machine or profile directory entry.

22.5.2 Edit the EXTENT CONTROL file

If you want to add disks to virtual machines, you should first let DirMaint know about them by updating the EXTENT CONTROL file. To do so, perform the following steps:

1. Log on to any SSI member as MAINT.
2. Invoke the command for DirMaint to send the current EXTENT CONTROL file:

```
==> dirm send extent control
DVHXMT1191I Your SEND request has been sent for processing to DIRMAINT
DVHXMT1191I at POKDEV62.
Ready; T=0.01/0.01 10:29:42
 DVHREQ2288I Your SEND request for MAINT at * has been accepted.
RDR FILE 0008 SENT FROM DIRMAINT PUN WAS 0049 RECS 0070 CPY  001 A NOHOLD
NOKEEP
 DVHREQ2289I Your SEND request for MAINT at * has completed; with RC = 0.
```

3. Receive the file from MAINT's reader:

```
==> receive 8 (rep
File EXTENT CONTROL A1 replaced by EXTENT CONTROL E1 received from DIRMAINT at
P
OKDEV62
```

4. Edit the file, adding disks to the REGIONS section. In the example that follows, two volumes are added, JM63AB and JM63AC, The RegionID, field 1, is set to the same value as the VolSer (label), field 2. Fields 3 and 4 set the cylinder range to all cylinders except cylinder 0, and the last field informs DirMaint as to the size of the disk:

```
==> x extent control
```

```
...
:REGIONS.
*RegionId  VolSer    RegStart      RegEnd  Dev-Type  Comments
JM61A5     JM61A5    0001          END     3390-09
JM61B2     JM61B2    0001          END     3390-09
JM6289     JM6289    0001          END     3390-03
JM628C     JM628C    0001          END     3390-03
JM628D     JM628D    0001          END     3390-03
JM628E     JM628E    0001          END     3390-03
JM628F     JM628F    0001          END     3390-03
JM6290     JM6290    0001          END     3390-03
JM6293     JM6293    0001          END     3390-03
JM6294     JM6294    0001          END     3390-03
JM6327     JM6327    0001          END     3390-03
JM6328     JM6328    0001          END     3390-03
JM6339     JM6339    0001          END     3390-03
JM633A     JM633A    0001          END     3390-03
JM633B     JM633B    0001          END     3390-03
JM633C     JM633C    0001          END     3390-03
:END.
:GROUPS.

JMD9AB    JMD9AB          1        10016  3390-09

:END.
:GROUPS.
...
```

5. Invoke the command for DirMaint to replace the current EXTENT CONTROL file:

```
==> dirm file extent control
PUN FILE 0009 SENT TO   DIRMAINT RDR AS  0053 RECS 0074 CPY  001 0 NOHOLD
NOKEEP
DVHXMT1191I Your FILE request has been sent for processing to DIRMAINT
DVHXMT1191I at POKDEV62.
Ready; T=0.01/0.01 10:32:38
 DVHREQ2288I Your FILE request for MAINT at * has been accepted.
 DVHRCV3821I File EXTENT CONTROL E1 has been received; RC = 0.
 DVHREQ2289I Your FILE request for MAINT at * has completed; with RC = 0.
```

Always watch for a return code of 0.

6. Invoke the command for DirMaint to reload the EXTENT CONTROL file in the current session:

```
==> dirm rlde
DVHXMT1191I Your RLDEXTN request has been sent for processing to
DVHXMT1191I DIRMAINT at POKDEV62.
Ready; T=0.01/0.01 10:34:04
 DVHREQ2288I Your RLDEXTN request for MAINT at * has been accepted.
 DVHILZ3510I Starting DVHINITL with directory: USER DIRECT E
 DVHILZ3510I DVHINITL Parms: BLDMONO BLDDASD BLDLINK
 DVHIZD3528W One or more DASD volume control files (CVD964) were
 DVHIZD3528W created using default values for device characteristics -
 DVHIZD3528W $ALLOC$ 0A00
 DVHREQ2289I Your RLDEXTN request for MAINT at * has completed; with RC =
 DVHREQ2289I 0.
```

Always watch for a return code of 0.

This section has shown how to update the DirMaint `EXTENT CONTROL` file.

22.5.3 Get a copy of the user directory

Sometimes it is convenient to get a copy of the entire user directory similar to editing the `USER DIRECT` file. To do so, perform the following steps:

1. Log on to any SSI member as `MAINT`.
2. Use the **USER WITHPASS** operand of DirMaint to send a copy of the user directory to `MAINT`'s reader:

```
==> dirm user withpass
DVHXMT1191I Your USER request has been sent for processing to DIRMAINT
DVHXMT1191I at POKDEV62.
Ready; T=0.01/0.01 10:57:20
 DVHREQ2288I Your USER request for MAINT at * has been accepted.
RDR FILE 0010 SENT FROM DIRMAINT PUN WAS 0057 RECS 5441 CPY  001 A NOHOLD
NOKEEP
 DVHREQ2289I Your USER request for MAINT at * has completed; with RC = 0
```

Watch for a 0 return code. Also, note the number of the reader file - **10** in this example.

3. Receive the file with the **REPLACE** option:

```
==> receive 10 (rep
File USER WITHPASS A0 replaced by USER WITHPASS A0 received from DIRMAINT at
POK
DEV62
```

4. Browse the user directory file, `USER WITHPASS A`.

```
==> browse user withpass
...
```

This section has shown how to get a copy of the current user directory.

22.5.4 Add an IDENTITY

Creating an IDENTITY (MCVM) is new to z/VM 6.3. Following is an example of defining a new MCVM that can be logged on to both SSI members at the same time:

1. Create a file with a file name corresponding to the `IDENTITY` and a file type of `DIRECT`. The `IDENTITY` is added without the `BUILD` statements because DirMaint will add the `BUILD` statements to the `IDENTITY` automatically when the `SUBCONFIG` is added:

```
==> x zmapvm62 direct
IDENTITY ZMAPVM62 DRCT 512M 1G BDEG
 INCLUDE LNXDFLT
 OPTION LNKNOPAS LANG AMENG
```

2. Add the `IDENTITY` to the user directory with the `ADD` operand:

```
==> dirm add zmapvm62
...
```

3. Create a file with a file name corresponding to the `SUBCONFIG` of the first member and a file type of `DIRECT`:

```
==> type zmapvm-1 direct
SUBCONFIG ZMAPVM-1
 MDISK 0100 3390 0001 10016 JM61BE MR LNX4VM LNX4VM LNX4VM
```

```
MDISK 0101 3390 0001 3338  JM6134 MR LNX4VM LNX4VM LNX4VM
MDISK 0102 3390 0001 3338  JM613F MR LNX4VM LNX4VM LNX4VM
MDISK 0103 3390 0001 3338  JM6140 MR LNX4VM LNX4VM LNX4VM
```

4. Add the SUBCONFIG to the user directory with the ADD operand and the following parameters:

   ```
   ==> dirm add zmapvm-1 build on ZVM63A in zmapvm62
   ...
   ```

5. Repeat the previous two steps for each additional member in the SSI.

This section has shown how to add an MCVM using DirMaint.

22.5.5 Clean up work units

Sometimes when you try to delete minidisks or entire virtual machines, there is an error because a disk to be deleted is linked or accessed. When this happens, failed work units can be the result. Before the deletion can be completed, the failed work units must be cleaned up.

To clean up failed work units, perform the following steps:

1. List all work units:

   ```
   ==> dirm status workunit all
   ```

2. List the specific work unit you want to delete. In this example, the work unit 20024211 is used:

   ```
   ==> dirm status workunit 20024211
   ```

3. Delete a specific work unit:

   ```
   ==> dirm workunit 20024211 cancel
   ```

Also, when you cannot get anywhere and want to start over, try the following command:

```
==> dirm for datamove cleanup cancel
```

22.5.6 Configure LogonBy processing

DirMaint can be configured to require users to log on with their own credentials. This is called *LogonBy processing*. This is required for a proper *audit trail* because it will allow SMF to capture each individual's access.

The function of LOGONBY is similar to the use of SURROGAT class profiles in z/OS. It is good practice that when a LOGONBY profile has been defined for a generic virtual machine, it is no longer possible to use the standard password to log on.

The following example will create userid1 and give it access to SYSADMIN:

1. Log on as MAINT.
2. Create a file called USERID1 DIRECT A with the following data:

   ```
   ==> x userid1 direct
   USER USERID1 PASSWORD1 512M 1G G
   ```

3. Issue the **DIRM ADD** command for that virtual machine:

   ```
   ==> dirm add userid1
   PUN FILE 0092 SENT TO   DIRMAINT RDR AS  0057 RECS 0011 CPY  001 0 NOHOLD
   NOKEEP
   ```

```
DVHXMT1191I Your ADD request has been sent for processing to DIRMAINT at
DVHXMT1191I ZVM63A.
Ready; T=0.01/0.01 09:36:19
 DVHREQ2288I Your ADD request for USERID1 at * has been accepted.
 DVHBIU3450I The source for directory entry USERID1 has been updated.
 DVHBIU3424I The next ONLINE will take place immediately.
 DVHDRC3451I The next ONLINE will take place via delta object directory.
 DVHRLA3891I Your DSATCTL request has been relayed for processing.
 DVHBIU3428I Changes made to directory entry USERID1 have been placed
 DVHBIU3428I online.
 DVHREQ2289I Your ADD request for USERID1 at * has completed; with RC
 DVHREQ2289I = 0.
 DVHREQ2288I Your DSATCTL request for DIRMAINT at
 DVHREQ2288I * has been accepted.
 DVHREQ2289I Your DSATCTL request for DIRMAINT at
 DVHREQ2289I * has completed; with RC = 0.
```

4. Set up the surrogate RACF class if it is not already created:

```
==> rac setr classact(surrogat)
==> rac setr generic(surrogat)
==> rac setr gencmd(surrogat)
==> rac setr classact(surrogat)
==> rac setr raclist(surrogat)
```

5. Only allow logon by processing for SYSADMIN:

```
==> rac rdef surrogat logonby.SYSADMIN audit(all)
```

6. Allow SYSADMIN to be logged on to by USERID1:

```
==> rac permit logonby.sysadmin cl(surr) acc(read) id(userid1)
==> rac setr raclist(surr) refresh
```

7. Test the logon:

```
z/VM ONLINE


                         / VV          VVV MM       MM
                        /  VV        VVV  MMM     MMM
                ZZZZZZ /   VV      VVV   MMMM   MMMM
                   ZZ /    VV    VVV    MM MM MM MM
                  ZZ /     VV  VVV     MM  MMM  MM
                 ZZ /      VVVVV      MM   M   MM
                ZZ /       VVV       MM       MM
               ZZZZZZ /    V        MM       MM

                 built on IBM Virtualization Technology




Fill in your USERID and PASSWORD and press ENTER
(Your password will not appear when you type it)
USERID   ===>
PASSWORD ===>

COMMAND  ===>  logon sysadmin by userid1
                                                          RUNNING   ZVM63A
```

8. You will be asked to change the password at first logon:

```
logon sysadmin by userid1

Enter your password,
or
To change your password, enter: ccc/nnn/nnn
     where ccc = current password, and nnn = new password

RPIMGR042I PASSWORD EXPIRED

To change your password - enter: nnn/nnn  where nnn = new password
or,
enter LOGOFF to cancel

 ICH70001I SYSADMIN LAST ACCESS AT 09:58:11 ON TUESDAY, JUNE 11, 2013
 HCPRPW004I Password changed
z/VM Version 6 Release 3.0, Service Level 0000 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES:   NO RDR,   NO PRT,   NO PUN
LOGON AT 10:10:58 EDT TUESDAY 06/11/13
z/VM V6.3.0    2013-06-04 12:50
```

You can issue a **QUERY USERID** command to see that you are logged on as SYSADMIN with its privileges.

22.5.7 Use the RACF SMF data unload utility

The *RACF SMF data unload utility* is a simple way to extract RACFs type 80 SMF data. The following example will show the TYPE 80 SMF record where USERID1 was created and given access to logon-by SYSADMIN. The virtual machine that will access the RACFADU EXEC will need RACF AUDITOR access. It will need to link to the RACFVM SMF output disks 301 and 302. The utility **RACFAU** is on RACFVM 305 disk.

1. Log on to MAINT.
2. Link the RACF 301, 302, and 305 disks:

```
==> link racfvm 301 301
==> link racfvm 302 302
==> link racfvm 305 305
==> acc 305 b
```

Note: To access the RACFVM 301 disk, you need to have RACF AUDITOR privileges.

3. Run the **RACFADU EXEC** using the 301 disk as input and the 191 disk as output.

Note: The RACFADU will work only if the output disk (191) is accessed as filemode A. In this example, the output file will be twice the size of your 301 used space.

```
==> RACFADU 301 191
   RACFADU  OUTPUT
     RPIADU033I SMF unload completed successfully.
     View the RACFADU MESSAGES file for additional details.
```

The output should now be in the `RACFADU OUTPUT A` file.

23

Monitor z/VM and Linux

> "*Not everything that can be counted counts, and not everything that counts can be counted.*"
>
> — Albert Einstein

This chapter briefly describes how to monitor z/VM and Linux. For another source on z/VM performance and monitoring, see Chapter 11, *Monitoring performance and capacity*, in the manual *Getting Started With Linux on System z*, SC24-6096, on the web at the following site:

`http://publib.boulder.ibm.com/cgi-bin/bookmgr/download/HCSXOC20.pdf?DT=20130528134905&XKS=hcsh2ac2`

There are a number of z/VM monitoring tools such as the CA VM:Monitor, IBM z/VM Performance Toolkit, IBM Tivoli® OMEGAMON® XE for z/VM and Linux, and products from IBM Velocity™ Software. The IBM z/VM Performance Toolkit is briefly described in this chapter.

This chapter contains the following sections:

23.1 Use basic z/VM commands

z/VM has many commands to monitor the state of the system. **CP INDICATE** is the most commonly used, and there are other commands that are addressed. For more information, see the *z/VM Performance Resources* web page at the following site:

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

23.1.1 Use the INDICATE command

z/VM has some basic commands, such as **INDICATE**. There are many **INDICATE** parameters that can be included as command-line options. Use the **HELP INDICATE** command 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 versions of this, depending on whether the issuing virtual machine has privilege class G or 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 virtual machines with class E privilege (for example, MAINT, OPERATOR) is shown here. The lines are numbered for clarity with a description that follows:

```
==> ind load
1  AVGPROC-000% 04
2  MDC READS-000068/SEC WRITES-000001/SEC HIT RATIO-099%
3  PAGING-0/SE
4  Q0-00001(00000)                           DORMANT-00012
5  Q1-00000(00000)           E1-00000(00000)
6  Q2-00001(00000) EXPAN-001 E2-00000(00000)
7  Q3-00001(00000) EXPAN-001 E3-00000(00000)
8
9  PROC 0000-000% CP   VM     PROC 0001-000% CP   VL
10 PROC 0002-000% IFL  VM     PROC 0003-000% IFL  VL
11
12 LIMITED-00000
```

The **INDICATE LOAD** command gives a snapshot of current system performance. Except for the counts of virtual machines in various queues and the limited list, the values you see here are a smoothed average over the past 4 minutes. The following areas are where z/VM performance analysts tend to focus:

- AVGPROC on line **1** gives the overall processor utilization, 38% in this example. The number following it is the number of online processors, 3 in this example. The individual processor utilization is shown on lines **9** and **10**. Take a glance at these to see if they are somewhat balanced. There are cases where an imbalance is okay. This would include very low utilization scenarios or cases where there are not enough users ready to run virtual processors to keep the physical processors busy. One of the processors will be a Master, all of the others Alternate, and some imbalance can result from performing these functions. Another imbalance will come also from vertical CPU management.
- (MDC) statistics are given on the second line. The effectiveness of MDC can be judged by the combination of the READS rate and the HIT RATIO. If both are high, many physical I/Os are avoided due to the MDC feature. For a system that has an appreciably high I/O rate, which is composed of reads plus writes, and a high proportion of reads, and a good hit ratio for those reads (tending to 90% or greater), the real, physical I/O avoidance can be very high. This author has seen the avoidance as high as 50% in some cases. Conversely, however, a high HIT RATIO with a low value for the READS rate should not be taken as good (100% hit ratio, when doing only 1 I/O per second is effectively meaningless).
- Line **3** describes more storage (memory) management. The PAGING rate is important. Higher values will often impact performance. This can be at least partially offset by increasing the number of page volumes, but a more thorough examination of this problem is advisable whenever it arises.
- On lines **4** through **7**, you also see a series of counters that represent the users in various queues. The z/VM scheduler classifies work into three different classes (1 through 3) and

a special additional class labeled *zero*. So the Column of Q_x values and E_x represent the virtual machines in the dispatch list and the eligible list. The most important value here to validate is that there are no virtual machines in the Eligible list: E1, E2, E3; this implies z/VM has stopped dispatching some virtual machines to avoid over committing resources. Such a system would require further investigation, possibly leading to some tuning work, or even hardware addition in extreme cases. 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
MAINT         Q1 R00   00001623/00001552   .I..  .0004
TCPIP         Q0 PS    00003496/00003178   .I..  99999
```

This is another class E command and displays the virtual processors associated with a given virtual machine (that can have multiple virtual processors) what queue (dispatch list, eligible list, limit list) they are in and what state they are in. This is a snapshot in time. Again, you want to check this output to ensure that there are no virtual machines in the eligible list. Normal virtual processors in the dispatch list will be Q_x (x=1,2,3). The eligible list would be marked as E_x. The third column in the example also gives state of virtual processor. This can be helpful to get an idea of how the virtual processors might be constrained. Virtual processors that are actually running in the snapshot period are marked with and RNN where NN is the processor number they are on. An R without a number means that the virtual processor is ready to run, but 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 does not have to be concerned about the other columns unless detailed analysis is required or if IBM support requests it. Also, always remember that is just a snapshot in time so often repeating this command over time can give a more accurate picture of your z/VM system. A single snapshot cannot be regarded as indicative.

23.1.2 Use other basic commands

Some other useful basic commands are briefly mentioned. All examples are shown from the MAINT virtual machine. The results will be different for users with fewer privileges.

Getting help

To get help on the system, use the **HELP** command. Sometimes it is hard to find help for exactly the command you are looking for. Some useful help commands are as follows:

```
==> help            // for basic help
==> help menus      // for menu of all z/VM help menus
==> help cp menu    // for a menu of all CP commands
==> help cpquery    // for a menu of all CP QUERY command
==> help cpset      // for a menu of all CP SET commands
```

Determining who is logged on

To see who is logged on to the system, use the **QUERY NAMES** command. For example:

```
==> q n
DIRMSAT2 - SSI
ZMAPVM62 - DSC , LINUX153 - DSC , LNXADMIN - DSC , LINUX157 - DSC
VSMEVSRV - DSC , VSMPROXY - DSC , VSMREQIU - DSC , VSMREQI6 - DSC
VSMREQIN - DSC , DTCSMAPI - DSC , PERSMAPI - DSC , VSMWORK3 - DSC
VSMWORK2 - DSC , VSMWORK1 - DSC , FTPSERVE - DSC , VSMGUARD - DSC
TCPIP    - DSC , DIRMAINT - DSC , DTCVSW2  - DSC , DTCVSW1  - DSC
VMSERVP  - DSC , VMSERVR  - DSC , VMSERVU  - DSC , VMSERVS  - DSC
```

```
OPERSYMP - DSC , DISKACNT - DSC , EREP     - DSC , OPERATOR - DSC
MAINT    -L0004
VSM     - TCPIP
```

Determining storage or memory

To see how much main storage (memory) is installed and allocated to a system, use the **QUERY STORAGE** command. For example:

```
==> q stor
STORAGE = 16G CONFIGURED = 16G INC = 256M STANDBY = 0  RESERVED = 0
```

This shows that there is 16 GB of central memory (storage).

Determining processors or CPUs

To see how many processors (CPs, IFLs, CPUs) you have allocated at system level, use the **QUERY PROCESSORS** command. For example:

```
==> q proc
PROCESSOR 00 MASTER CP
PROCESSOR 01 ALTERNATE CP
PROCESSOR 02 ALTERNATE CP
PROCESSOR 03 ALTERNATE CP
PROCESSOR 04 ALTERNATE CP
PROCESSOR 05 ALTERNATE CP
PROCESSOR 06 ALTERNATE CP
PROCESSOR 07 ALTERNATE CP
PROCESSOR 08 ALTERNATE CP
PROCESSOR 09 ALTERNATE CP
```

Determining software level

To determine what level of CP your system is at, use the **QUERY CPLEVEL** command. For example:

```
==> q cplevel
z/VM Version 6 Release 3.0, service level 1301 (64-bit)
Generated at 06/28/13 14:58:28 EDT
IPL at 09/04/13 10:48:34 EDT
```

Determining system cylinder allocation

The **QUERY ALLOC MAP** command shows you the system allocation of spool, paging, and directory space. For example:

```
==> q alloc map
                EXTENT     EXTENT                           % ALLOCATION
VOLID  RDEV      START        END  TOTAL IN USE   HIGH USED TYPE
------ ---- ---------- ---------- ------ ------ ------ ---- -------------
JV1030 1030          1         20     20      1      1   5% DRCT ACTIVE
JV1031 1031          1       3338 600840  87022  91029  14% SPOOL
JV1131 1131          -          -      0      0      0   0  SHARED
JP1260 1260          0      10016  1761K     27     56   1% PAGE
JP1261 1261          0      10016  1761K     75     75   1% PAGE
JV1032 1032          1       3338 600840     52     63   1% PAGE
```

Determining DASD, OSA, and virtual resources

The **QUERY DASD** and **QUERY DASD FREE** commands 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. The following list gives the short form of these commands without any of the associated output shown:

```
==> q da
==> q da free
==> q osa
==> q osa free
==> q v all
```

23.2 The z/VM Performance Toolkit

To use the z/VM Performance Toolkit, the product must be ordered. You should configure the product only if you ordered it. z/VM Performance Toolkit is part of the z/VM base installation and it is installed as disabled. It is a priced feature of z/VM.

Much more detail can be found in the following books:

- *z/VM Performance Toolkit Guide*, SC24-6156, *z/VM Performance Toolkit Reference*, SC24-6157, on the web starting at the z/VM 6.3 bookshelf:

  http://publib.boulder.ibm.com/cgi-bin/bookmgr/XKS/hcsh2ac2

  Search for **Toolkit** on that page.
- *The Program Directory for Performance Toolkit for VM*, GI10-0785-00:

  http://www.vm.ibm.com/progdir/6vmptk30.pdf
- The IBM Redbooks publication *Linux on IBM zSeries and S/390®: Performance Toolkit for VM*, SG24-6059, on the web at the following site:

  http://www.redbooks.ibm.com/abstracts/sg246059.html

The section that follows describes how to set up and use the IBM Performance Toolkit *very briefly*:

- "Configure the IBM Performance Toolkit for VM" on page 421
- "Use the IBM Performance Toolkit for VM" on page 426

23.2.1 Configure the IBM Performance Toolkit for VM

The Performance Toolkit is installed with z/VM. Configuration is described in the Program Directory for Performance Toolkit for VM, which can be found at the following website:

http://www.ibm.com/eserver/zseries/zvm/library

The following is a summary of how to turn it on. Again, you should configure the product only if you ordered it:

1. Query which priced products are enabled with the **QUERY PRODUCT** command:

   ```
   ==> q product
   Product  State    Description
   IBMVMSSI Enabled  IBM z/VM Single System Image Feature
   6VMDIR30 Disabled 00/00/00.00:00:00.$BASEDDR DIRECTORY MAINTENANCE FACILITY (Dir
   Maint)
   6VMPTK30 Disabled 00/00/00.00:00:00.$BASEDDR PERFORMANCE TOOLKIT FOR VM
   6VMRAC30 Disabled 00/00/00.00:00:00.$BASEDDR RACF Security Server
   6VMRSC30 Disabled 00/00/00.00:00:00.$BASEDDR RSCS Networking
   ```
2. To enable IBM Performance Toolkit for VM, log on to **MAINT630** and enter the following command:

   ```
   ==> service perftk enable
   VMFSRV2760I SERVICE processing started
   ```

```
...
VMFSUT2760I VMFSUFTB processing started
VMFSUT2760I VMFSUFTB processing completed successfully
VMFSRV2760I SERVICE processing completed successfully
```

You should see a few screens of messages scroll by and finally the success messages shown above. This will enable Performance Toolkit for the current z/VM session.

3. The `SYSTEM CONFIG` file is modified by having a line appended to the end. Verify this has been added with the following commands:

```
==> vmlink pmaint cf0
DMSVML2060I PMAINT CF0 linked as 0120 file mode Z
==> type system config z
...    // many screens cleared
PRODUCT PRODID 6VMPTK30 STATE ENABLED DESCRIPTION '06/05/13.15:22:55.MAINT630 PE
RFKIT Minidisk Install and Service'
```

4. The **QUERY PRODUCT** command will show the change:

```
==> q product
Product  State    Description
IBMVMSSI Enabled  IBM z/VM Single System Image Feature
6VMDIR30 Disabled 00/00/00.00:00:00.$BASEDDR DIRECTORY MAINTENANCE FACILITY (Dir
Maint)
6VMPTK30 Enabled  06/05/13.15:22:55.MAINT630 PERFKIT Minidisk Install and Servic
e
6VMRAC30 Disabled 00/00/00.00:00:00.$BASEDDR RACF Security Server
6VMRSC30 Disabled 00/00/00.00:00:00.$BASEDDR RSCS Networking
```

The Performance Toolkit is now enabled. You can also verify by running the **QUERY PRODUCT** command again.

23.2.2 Configure web browser support

After the product is enabled, the TCP/IP profile must be modified to enable web access to the Performance Toolkit. The following example sets the port to 80, which is the default for a web browser:

1. Log On to `TCPMAINT`. Edit the TCPIP configuration file - the default name is `PROFILE TCPIP`) and search for the string reserve ports. This is where z/VM TCP/IP ports are reserved:

```
==> x profile tcpip d
====> /port
```

2. Add the following line under the PORT entries:

```
...
PORT
  20   TCP FTPSERVE  NOAUTOLOG ; FTP Server
  21   TCP FTPSERVE            ; FTP Server
  23   TCP INTCLIEN            ; TELNET Server
; 25   TCP SMTP                ; SMTP Server
  80   TCP PERFSVM             ; Performance Toolkit
; 111  TCP PORTMAP             ; Portmap Server
; 111  UDP PORTMAP             ; Portmap Server
; 143  TCP IMAP                ; IMAP Server
...
```

Save your changes.

3. To change TCP/IP dynamically, use the **OBEYFILE** command:

```
==> netstat obey port 80 tcp perfsvm
VM TCP/IP Netstat Level 630       TCP/IP Server Name: TCPIP
```

```
OBEY command response is: OK
OBEY return code = 0
```

4. Issue the **NETSTAT CLIENTS** command to verify your configuration. You want to see that the service named PERFSVM is a client. This should be shown after a few screens of output:

```
==> netstat clients
...
Client: PERFSVM                 Authorization: {none}
Notes Handled: none
Last Touched:   0:03:23
Vmcf error count: 0
```

If you are configuring central monitoring in a single system image (SSI) cluster, it is enough to configure the web server on only one of the members. Central monitoring enables one member to monitor the other members of the SSI cluster.

23.2.3 Configure PERFSVM

The PERFSVM virtual machine is the Performance Toolkit service machine. Follow these steps to configure it.

1. Log On to PERFSVM. If you successfully enabled the product, you will enter a Performance Toolkit session and see the following text at the top of the screen:

```
FCX001                Performance Toolkit for VM                Autoscroll 12
 FCXBAS500I Performance Toolkit for VM FL630
 16:14:15 Monitor event started -- recording is activated
 16:14:15 Monitor sample started -- recording is activated
```

2. Press **F12 twice** to get to a CMS prompt.
3. Copy the default configuration files, which are on PERFSVM's D disk, to your A disk:

```
==> copy * * d = = a
```

4. The main configuration file is FCONX $PROFILE. Edit that file and search for the string VMCF:

```
==> x fconx $profile
====> /vmcf
```

This should take you to line 190 where the next eight lines are comments starting with an asterisk (*****). Perform the following changes:

– Uncomment the 2nd, 4th, 6th, and 8th lines by changing *C to **FC**
– Change port 81 to **80** on the fourth line. This enables you to use a browser interface without having to specify port 81 on the URL (with a :81 suffix).

The modified lines should be as follows. Save your changes with the **FILE** subcommand:

```
*    Following command activates VMCF data retrieval interface
FC MONCOLL VMCF ON
*    Define the maximum allowed number of Internet connections
FC MONCOLL WEBSERV MAXCONN 100
*    Define the timeout of inactive Internet connections in minutes
FC MONCOLL WEBSERV TIMEOUT 30
*    Following command activates Internet interface
FC MONCOLL WEBSERV ON TCPIP TCPIP 80
*    Following command activates Internet interface with SSL
...
====> file
```

If you are configuring central monitoring in an SSI cluster, enable the four FC commands only on one member, which will serve as a web server. On the other members, allow only the first FC statement (**FC MONCOLL VMCF ON**)

5. Create a remote data retrieval authorization file with your z/VM system identifier (replace ***ZVM63A*** with your system identifier):

```
==> x fconrmt authoriz
====> a 2
ZVM63A PERFSVM  S&FSERV DATA
```

If you are configuring central monitoring in an SSI cluster, allow the member that serves as the web server to access the other members. The authorization file on a second member would look like what is shown below:

```
ZVM63A PERFSVM  DATA
ZVM63B PERFSVM  S&FSERV DATA
```

6. Create a system identification file that links your z/VM systems and PERFSVM to a special resource name. (replace ***ZVM63A*** with your system identifier):

```
==> x fconrmt systems
====> a
ZVM63A PERFSVM  z/VM6.3 N FCXC1R01
```

If you are configuring central monitoring in an SSI cluster, specify all other members as well. Make sure each uses a unique resource name. For example, the first member might be **FCXC1R01** and the second member **FCXC1R02**, and so on.

```
ZVM63A PERFSVM  z/VM6.3 N FCXC1R01
ZVM63B PERFSVM  z/VM6.3 N FCXC1R02
ZVM63C PERFSVM  z/VM6.3 N FCXC1R03
ZVM63D PERFSVM  z/VM6.3 N FCXC1R04
```

System identification files on all members must be the same.

7. Set up a resource override for the default resource name (enter the resource name that you used in FCONRMT AUTHORIZ):

```
==> x ucomdir names
====> a 6
:nick.FCXRES00 :luname.*IDENT
            :tpn.FCXC1R01
            :security.SAME
:nick.FCXSYSTM :luname.*IDENT
            :tpn.FCXC1S01
            :security.SAME
```

If you are configuring central monitoring in an SSI cluster, specify resource override on each member. The second member will use **FCXC1R02** and **FCXC1S02**, the third member **FCXC1R03** and **FCXC1S03** and the fourth member will use **FCXC1R04** and **FCXC1S04**.

8. Make CP start collecting performance data and start Performance Toolkit automatically after IPL:

   a. Log on to AUTOLOG1.

   b. Before pressing Enter at the VM READ prompt, type **acc (noprof** so that the **PROFILE EXEC** is not run:

```
LOGON AUTOLOG1
z/VM Version 6 Release 3.0, Service Level 0000 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES:   NO RDR, 0008 PRT,   NO PUN
LOGON AT 12:13:55 EDT THURSDAY 06/06/13
```

```
z/VM V6.3.0    2013-06-04 12:50
acc (noprof
Ready; T=0.01/0.01 12:14:01
```

c. Edit the profile exec in the following way:

```
==> x profile exec a

...
/**********************************************************************/
/* Customer processing can be added here                              */
/**********************************************************************/
"CP XAUTOLOG TCPIP"             /* Autolog TCPIP                      */
"CP SET MDC STOR OM 256M"       /* Limit minidisk cache in CSTOR      */
"CP SET SIGNAL SHUTDOWN 600"    /* Allow guests 10 min to shut down   */
"CP XAUTOLOG LNXADMIN"          /* Start the Linux admin machine      */

"CP MONITOR SAMPLE ENABLE PROCESSOR" /* Setup CP MONITOR parameters  */
"CP MONITOR SAMPLE ENABLE STORAGE"
"CP MONITOR SAMPLE ENABLE USER ALL"
"CP MONITOR SAMPLE ENABLE I/O ALL"
"CP MONITOR SAMPLE ENABLE NETWORK"
"CP MONITOR SAMPLE ENABLE APPLDATA ALL"
"CP MONITOR SAMPLE ENABLE ISFC"
"CP MONITOR SAMPLE ENABLE SSI"

"CP MONITOR EVENT  ENABLE STORAGE"
"CP MONITOR EVENT  ENABLE I/O ALL"
"CP MONITOR EVENT  ENABLE NETWORK"
"CP MONITOR EVENT  ENABLE ISFC"
"CP MONITOR EVENT  ENABLE SSI"

"CP MONITOR SAMPLE INTERVAL 1 MIN"  /* Set sampling interval */

"CP XAUTOLOG PERFSVM"           /* Start Performance Toolkit          */
```

d. Save the file using the following command:

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

> **Note:** If you do not plan to IPL before you try Performance Toolkit, you should run all CP MONITOR commands you just added to the **PROFILE EXEC** file so that CP starts to collect performance data.

e. Log off from AUTOLOG1.

23.2.4 Start the IBM Performance Toolkit for VM

To start the Performance Toolkit, perform the following steps:

1. Log on to the PERFSVM virtual machine.
2. Press **Enter** and the performance toolkit should start through the PROFILE EXEC:

```
FFCX001                Performance Toolkit for VM              Autoscroll 12
 FCXBAS500I Performance Toolkit for VM FL630
 12:32:15 FCXAPP530I Connected to *IDENT for resource FCXC1R01
 12:32:15 FCXAPF530I Connected to *IDENT for resource FCXC1S01
 12:32:15 FCXTCP571I Connected to TCP/IP server TCPIP on path 0003
```

```
12:32:15 FCXAPP527I User PERFSVM connected on path 0006
12:32:15 FCXAPC535I Connected to resource FCXC1R01 on path 0005, for S&F-Coll
12:32:15 FCXTCP575I WebServer host IP address is 9.12.7.11:00080
12:32:15 FCXTCP590I WebServer interface activated
12:32:15 Monitor event started -- recording is activated
12:32:15 Monitor sample started -- recording is activated
```

Disconnect from `PERFSVM` now.

```
Command ===> disc
```

The Performance Toolkit should now be configured and running.

23.2.5 Use the IBM Performance Toolkit for VM

The Performance Toolkit can be used with a web browser or 3270 interface.

Using a web browser interface

To use the web-enabled Performance Toolkit, perform the following steps:

1. Point a browser to your z/VM system. For example:

   `http://9.12.7.11`

2. You should see a splash screen, then the Web Server Logon window, as shown in Figure 23-1.

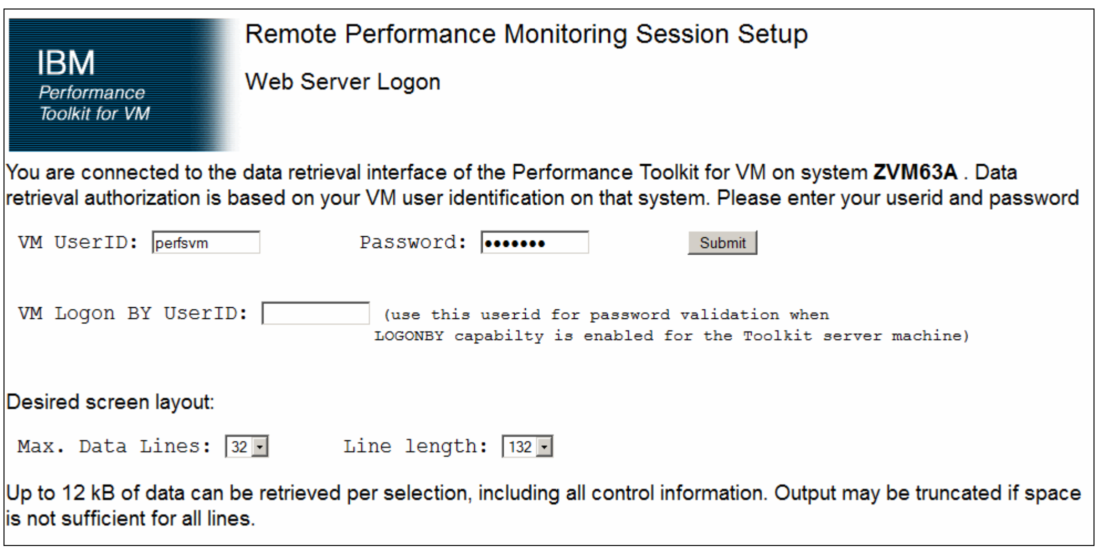


Figure 23-1 Performance Toolkit logon window

3. Enter any valid user ID and password and click **Submit**. In this example `PERFSVM` is used.
4. The *Central Monitoring System Load Overview* appears with your system identifiers (*Node-ID*) on the left side.
5. Click your system identifier and the *Initial Performance Data Selection Menu* window appears as shown in Figure 23-2 on page 427.
6. From this window, you can drill down into many different types of reports.

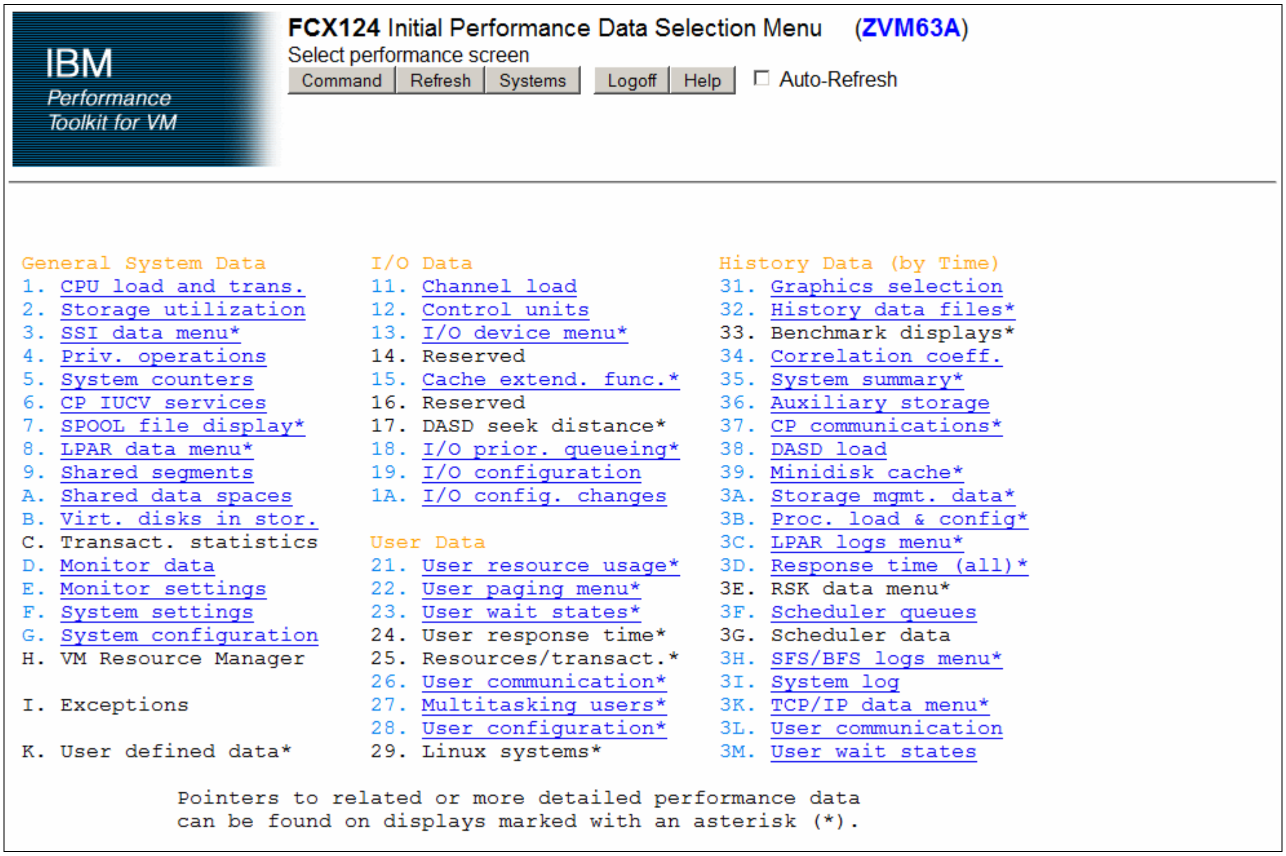


Figure 23-2 Browser interface to the Performance Toolkit

Use a 3270 interface

You can also use a 3270 interface. To do so, perform the following steps:

1. Log on to `PERFSVM`.
2. If you had disconnected, pressing Enter should get you back to the Performance Toolkit command line. If the virtual machine was logged off, the **`PROFILE EXEC`** should run and get you to the command line. Enter the **`MONITOR`** command:

   ```
   Command ==> monitor
   ```

   The Performance Screen Selection panel then appears, as shown in Example 23-1.

Example 23-1 Performance Screen Selection panel

```
FCX124          Performance Screen Selection  (FL630        )     Perf. Monitor

General System Data         I/O Data                     History Data (by Time)
1. CPU load and trans.      11. Channel load             31. Graphics selection
2. Storage utilization      12. Control units            32. History data files*
3. SSI data menu*           13. I/O device menu*         33. Benchmark displays*
4. Priv. operations         14. Reserved                 34. Correlation coeff.
5. System counters          15. Cache extend. func.*     35. System summary*
6. CP IUCV services         16. Reserved                 36. Auxiliary storage
7. SPOOL file display*      17. DASD seek distance*      37. CP communications*
8. LPAR data menu*          18. I/O prior. queueing*     38. DASD load
9. Shared segments          19. I/O configuration        39. Minidisk cache*
A. Shared data spaces       1A. I/O config. changes      3A. Storage mgmt. data*
B. Virt. disks in stor.                                  3B. Proc. load & config*
C. Transact. statistics     User Data                    3C. LPAR logs menu*
```

```
D. Monitor data            21. User resource usage*   3D. Response time (all)*
E. Monitor settings        22. User paging menu*      3E. RSK data menu*
F. System settings         23. User wait states*      3F. Scheduler queues
G. System configuration    24. User response time*    3G. Scheduler data
H. VM Resource Manager     25. Resources/transact.*   3H. SFS/BFS logs menu*
                           26. User communication*    3I. System log
I. Exceptions              27. Multitasking users*    3K. TCP/IP data menu*
                           28. User configuration*    3L. User communication
K. User defined data*      29. Linux systems*         3M. User wait states

            Pointers to related or more detailed performance data
            can be found on displays marked with an asterisk (*).
```

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
33. Benchmark displays
```

23.3 Collect and use raw CP monitor data

While the Performance Toolkit formats and displays current performance data, it is often needed to look at older data as well. Typical use would be to compare the current system performance to the past performance to have data available for troubleshooting, or to generate reports.

23.3.1 Collect CP monitor data

CP monitor records are collected by **MONWRITE** utility and written to a disk or tape. Resulting file contains all the original unprocessed data. This data can be later used to generate reports or the Performance Toolkit can use it in Monitor Data Scan Mode to look at historical data as if it was current:

1. Log on to the **MONWRITE** virtual machine.
2. Edit the **PROFILE EXEC**:

```
LOGON MONWRITE
z/VM Version 6 Release 3.0, Service Level 0000 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES:   NO RDR,   NO PRT,   NO PUN
LOGON AT 10:40:31 EDT FRIDAY 06/07/13
z/VM V6.3.0    2013-06-04 12:50

Ready; T=0.01/0.01 10:40:34
==> x profile exec a
input
/* ALL MONITOR COMMANDS ARE LOCATED IN AUTOLOG1'S PROFILE EXEC */
```

```
'MONWRITE MONDCSS *MONITOR DISK CLOSE 480'
==> file
```

3. Execute the REXX exec named **profile**

```
==> profile
HCPMOW6272I Now recording in file D060713 T110146 A1
HCPMOW6265A MONITOR WRITER CONNECTED TO *MONITOR
```

4. Disconnect from MONWRITE

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

The **CLOSE 480** statement tells **MONWRITE** to close the output file every 8 hours (480 minutes) starting from midnight. It means, regardless of when it starts recording, it will close the file at 08:00, 16:00 and at 24:00. The file name will clearly show the date and time when recording started.

To collect **MONWRITE** data automatically, start the **MONWRITE** virtual machine when z/VM is IPLed. To do so, add a line to the **PROFILE EXEC** of the AUTOLOG1 191 disk (or AUTOLOG2 191 if an external security manager such as RACF is running).

```
==> x profile exec
...
"CP XAUTOLOG MONWRITE"              /* Start the MONWRITE VM       */
...
```

MONWRITE's A-disk is shipped as 300 cylinders, which is quite small. Depending on monitor interval activity of the system and the number of samples/events it can fill up very quickly. When the disk is full, **MONWRITE** will not be able to write anymore. It is important to monitor the space on **MONWRITE's** A-disk. Another possibility is to use some utility that would archive old files and clean up the space automatically. An example of such a utility would be MONCLEAN. You can download it from the following site:

http://www.vm.ibm.com/download/packages/descript.cgi?MONCLEAN

Follow these steps for the MONCLEAN installation:

1. Use FTP binary to transfer MONCLEAN VMARC to MONWRITE's 191 disk.
2. Run MONWRITE VMARC via pipe command:

```
==> pipe < monclean vmarc a | fblock 80 00 | > monclean vmarc A F 80
```

3. Unpack the MONCLEAN VMARC file with the **VMARC** command:

```
==> vmarc unpk monclean vmarc a
MONCLEAN EXEC     A1. Bytes in=      4080, bytes out=      7678 (   188%).
MONCLEAN README   A1. Bytes in=      1040, bytes out=      2240 (   215%).
```

4. Check documentation in MONCLEAN README.
5. Modify PROFILE EXEC:

```
==> x profile exec
/* ALL MONITOR COMMANDS ARE LOCATED IN AUTOLOG1'S PROFILE EXEC */
'MONWRITE MONDCSS *MONITOR DISK CLOSE 60 EXEC MONCLEAN'
```

6. Start recording:

```
==> profile
HCPMOW6272I Now recording in file D061213 T131724 A1
HCPMOW6265A MONITOR WRITER CONNECTED TO *MONITOR
```

7. MONWRITE will close the output file every hour and execute MONCLEAN EXEC. If the MONCLEAN EXEC was not modified, it will remove the oldest file when the disk reaches 80% full.

8. Example 23-2 shows `MONWRITE's` 191 disk when **MONCLEAN** is running.

Example 23-2 MONWRITE 191 disk

```
MAINT    FILELIST A0  V 169  Trunc=169 Size=19 Line=1 Col=1 Alt=0
Cmd   Filename Filetype Fm Format Lrecl    Records     Blocks   Date     Time
      D061313  T100016  Z1 F       4096      49275      49275  6/13/13 10:29:16
      D061313  T090016  Z1 F       4096      99407      99407  6/13/13 10:00:15
      D061313  T080015  Z1 F       4096      99392      99392  6/13/13  9:00:15
      D061313  T070015  Z1 F       4096      99348      99348  6/13/13  8:00:15
      D061313  T060015  Z1 F       4096      99348      99348  6/13/13  7:00:15
      D061313  T050016  Z1 F       4096      99348      99348  6/13/13  6:00:15
      D061313  T040016  Z1 F       4096      99348      99348  6/13/13  5:00:15
      D061313  T030015  Z1 F       4096      99348      99348  6/13/13  4:00:15
      D061313  T020016  Z1 F       4096      99348      99348  6/13/13  3:00:15
      D061313  T010015  Z1 F       4096      99348      99348  6/13/13  2:00:15
      D061313  T000015  Z1 F       4096      99348      99348  6/13/13  1:00:15
      D061213  T230015  Z1 F       4096      99348      99348  6/13/13  0:00:15
      D061213  T220015  Z1 F       4096      99356      99356  6/12/13 23:00:15
      D061213  T210015  Z1 F       4096      99357      99357  6/12/13 22:00:15
      D061213  T200015  Z1 F       4096      99348      99348  6/12/13 21:00:15
      PROFILE  EXEC     Z1 V         65          2          1  6/12/13 11:35:49
      MONCLEAN EXEC     Z1 V         75        194          2  6/12/13 11:32:13
      MONCLEAN README   Z1 F         80         28          1  6/12/13 11:32:13
      MONCLEAN VMARC    Z1 F         80         64          2  6/12/13 11:32:13
```

23.3.2 Use CP monitor data

The Performance Toolkit subcommand **MONSCAN** allows you to select a CP monitor file on disk or tape (created by the standard **MONWRITE** utility) as input for performance data analysis. When the specified file is found, a performance data scan mode is entered which looks almost identical to the normal real-time monitoring mode and, which allows you to browse through the accumulated monitor data.

As `PERFSVM` virtual machine is used to show the current performance data, it is better to use a different virtual machine to perform **MONSCAN**. The following example uses the `MAINT` user ID:

1. Link and access `PERFSVM`'s 201 minidisk:

```
==> vmlink perfsvm 201
DMSVML2060I PERFSVM 201 linked as 0120 file mode Z
```

2. Link and access `MONWRITE`'s 191 minidisk:

```
==> vmlink monwrite 191
DMSVML2060I MONWRITE 191 linked as 0121 file mode X
```

3. Check files available from MONWRITE:

```
==> filel * * x
MAINT    FILELIST A0  V 169  Trunc=169 Size=4 Line=1 Col=1 Alt=0
Cmd   Filename Filetype Fm Format Lrecl    Records     Blocks   Date     Time
      D061013  T084824  X1 F       4096      53930      53930  6/10/13  9:20:43
      PROFILE  EXEC     X1 V         65          3          1  6/10/13  8:48:21
```

4. Run the MONSCAN subcommand:

```
==> perfkit monscan D061013 T084824 X
```

Regular Performance Screen Selection appears.

```
FCX124         Performance Screen Selection  (FL630       )    Monitor Scan
```

```
General System Data       I/O Data                  History Data (by Time)
1. CPU load and trans.    11. Channel load          31. Graphics selection
2. Storage utilization    12. Control units         32. History data files*
3. SSI data menu*         13. I/O device menu*      33. Benchmark displays*
4. Priv. operations       14. Reserved              34. Correlation coeff.
5. System counters        15. Cache extend. func.*  35. System summary*
6. CP IUCV services       16. Reserved              36. Auxiliary storage
7. SPOOL file display*    17. DASD seek distance*   37. CP communications*
8. LPAR data              18. I/O prior. queueing*  38. DASD load
9. Shared segments        19. I/O configuration     39. Minidisk cache*
A. Shared data spaces     1A. I/O config. changes   3A. Storage mgmt. data*
B. Virt. disks in stor.                             3B. Proc. load & config*
C. Transact. statistics   User Data                 3C. Logical part. load
D. Monitor data           21. User resource usage*  3D. Response time (all)*
E. Monitor settings       22. User paging load*     3E. RSK data menu*
F. System settings        23. User wait states*     3F. Scheduler queues
G. System configuration   24. User response time*   3G. Scheduler data
H. VM Resource Manager    25. Resources/transact.*  3H. SFS/BFS logs menu*
                          26. User communication*   3I. System log
I. Exceptions             27. Multitasking users*   3K. TCP/IP data menu*
                          28. User configuration*   3L. User communication
K. User defined data*     29. Linux systems*        3M. User wait states

            Pointers to related or more detailed performance data
            can be found on displays marked with an asterisk (*).
```

Make a selection, for example 1 - CPU Load. The first screen will not contain any data. Enter command **nexts** (next sample) and a screen with real numbers will appear. You can see the interval on the top of the screen.

```
FCX100      Data for 2013/06/10  Interval 08:48:40 - 08:49:40    Monitor Scan

CPU Load                                           Status or
PROC TYPE %CPU  %CP %EMU %WT %SYS %SP %SIC %LOGLD  ded. User
P00  CP      0    0    0 100    0   0   99      0  Master
P01  CP      0    0    0 100    0   0   99      0  Alternate
P02  IFL     0    0    0 100    0   0  ...      0  Alternate
P03  IFL     0    0    0 100    0   0  ...      0  Alternate

Total SSCH/RSCH    254/s     Page rate        .0/s     Priv. instruct.    28/s
Virtual I/O rate    10/s     XSTORE paging    .0/s     Diagnose instr.    16/s
Total rel. SHARE    3050     Tot. abs SHARE     0%

Queue Statistics:     Q0     Q1     Q2     Q3     User Status:
VMDBKs in queue        1      0      1      0     # of logged on users      14
VMDBKs loading         0      0      0      0     # of dialed users          0
Eligible VMDBKs               0      0      0     # of active users          7
El. VMDBKs loading            0      0      0     # of in-queue users        2
Tot. WS (pages)     2911      0  41870      0     % in-Q users in PGWAIT     0
Reserved                                          % in-Q users in IOWAIT     0
85% elapsed time   96.00  16.00  128.0  768.0     % elig. (resource wait)    0

Transactions      Q-Disp    trivial   non-trv     User Extremes:
Average users        2.7         .8        .2     Max. CPU %   LNXADMIN     .1
Trans. per sec.       .2         .1        .0     Reserved
Av. time (sec)     18.40      12.39     16.39     Max. IO/sec  MONWRITE    9.4
UP trans. time                 .000      .000     Max. PGS/s   ........  .....
MP trans. time                12.39     16.39     Max. RESPG   LNXADMIN  41923
System ITR (trans. per sec. tot. CPU)    31.3     Max. MDCIO   MONWRITE     .1
Emul. ITR (trans. per sec. emul. CPU)   269.2     Max. XSTORE  ........  .....
```

23.4 Monitor Linux performance for troubleshooting

Previous sections described how the Performance Toolkit can show resource consumption of the Linux guest as measured and dispatched by the z/VM hypervisor. z/VM is not aware of the nature of the guest and it cannot understand what is happening inside the guest. For that reason, it is important to have the ability to measure performance data from within the Linux guest itself. To monitor Linux performance data at this level, a data gatherer process must be running within each Linux guest that you want to monitor. There are different ways of gathering this data. There are many commercial and non-commercial solutions for long-term monitoring as well. This book cannot cover all the requirements for long-term monitoring (low CPU consumption, data storage, and similar). This chapter shows how to monitor Linux performance in short periods, especially when troubleshooting performance problems.

23.4.1 Monitor Linux performance from z/VM

This section describes how to gather Linux performance data in Linux and provide this data to z/VM for a consolidated overview.

To monitor Linux performance data directly from the kernel, the following statements must be true:

1. The `APPLMON` option must be set in the user directory.
2. Applmon data monitoring must be built into the kernel.

The first requirement should be true as the `OPTION APPLMON` was set for the Linux virtual machines in earlier sections. For the second requirement, both RHEL 6.4 and SLES11 SP3 have this feature built in.

A quick description of how to use this built-in monitoring function follows:

1. Start an SSH session to a Linux system. In this example, `LINUX3` is used.
2. 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
   ```

3. There is no output so no modules with the string `appldata` are loaded. Load those modules with the **modprobe** command and verify that they have been loaded:

   ```
   # modprobe appldata_mem
   # modprobe appldata_os
   # modprobe appldata_net_sum
   ```

4. Now if you repeat the **lsmod** command, you should see the following:

   ```
   # lsmod | grep appldata
   appldata_net_sum       1966  0
   appldata_os            2989  0
   appldata_mem           2008  0
   ```

5. The directory in the virtual `/proc/` file system where the monitoring variables exist is `/proc/sys/appldata/`. In this directory, there are five files as follows:

`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

6. To turn on the built-in kernel monitoring, use the **echo** command to send a non-zero value into four of the five monitoring variables in the /proc/ virtual file system:

```
# echo 1 > /proc/sys/appldata/timer
# echo 1 > /proc/sys/appldata/mem
# echo 1 > /proc/sys/appldata/os
# echo 1 > /proc/sys/appldata/net_sum
```

Built-in kernel monitoring should now be turned on. You might want to leave only 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.

View performance data from the Linux kernel in the Performance Toolkit

After the system has had some time to collect data, you should be able to use the Performance Toolkit to view Linux performance data. To view that data, drill down into menu 29, `Linux systems`. This can be done either from the browser interface or the 3270 interface, as shown in Example 23-3.

Example 23-3 Linux Displays

```
FCX242      CPU 2817  SER 23BD5          Linux Displays         Perf. Monitor



   Linux screens selection
 S Display     Description
 . LINUX       RMF PM system selection menu
 . LXCPU       Summary CPU activity display
 S LXMEM       Summary memory util. & activity display
 . LXNETWRK    Summary network activity display
```

Then, type **S** over the period on the left side of the submenu screen in the row corresponding to the report you want to see. You should see a new report screen with the Linux guest systems memory overview, as shown in Example 23-4

Example 23-4 Linux guests memory overview

```
FCX244      CPU 2817  SER 23BD5       Initial  14:22:57         Perf. Monitor
______            .      .       .      .      .       .      .       .      .
          <------------ Memory Allocation (MB) -------------> <------- Swapping
Linux     <--- Main ---> <--- High --->        Buffers  Cache <-Space (MB)-> <-
Userid    M_Total %MUsed H_Total %HUsed Shared /CaFree   Used S_Total %SUsed
>System<    491.6   25.8      .0     .0     .0     8.6   46.3   761.6     .0
LINUX3      491.6   25.8      .0     .0     .0     8.6   46.3   761.6     .0
```

You can also use a web interface to view the same data.

23.4.2 Monitor Linux performance from inside Linux

There are many tools for Linux performance monitoring. This section describes just some commonly used. They are all platform independent and they work on Linux in general.

top command

When running the **top** command without any parameters, it shows a system overview and running tasks similar to Example 23-5. Output is refreshed every 3 seconds automatically. To leave **top**, press “q”.

Example 23-5 top command

```
top - 11:49:13 up 29 min,  1 user,  load average: 0.95, 0.46, 0.18
Tasks:  90 total,   2 running,  88 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.9%us,  4.7%sy,  0.0%ni, 94.2%id,  0.1%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:    503392k total,   142544k used,   360848k free,    10100k buffers
Swap:   779872k total,        0k used,   779872k free,    56560k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 1687 root      20   0 99044  564  456 R 99.4  0.1   3:10.84 cat
    1 root      20   0  3248 1632 1312 S  0.0  0.3   0:00.29 init
    2 root      20   0     0    0    0 S  0.0  0.0   0:00.00 kthreadd
...
```

vmstat command

Another useful command is **vmstat**, which reports information about processes, memory, paging, block IO, traps, and cpu activity. When running **vmstat** without any parameters, it shows just one line summarizing averages since the last IPL, which is not very useful. Example 23-6 shows **vmstat 5** output. This shows the first line with averages since the last IPL and then writes a new line every 5 seconds with the current data.

Example 23-6 vmstat

```
[root@virtcook3 ~]# vmstat 5
procs -----------memory---------- ---swap-- -----io---- --system-- -----cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 1  0      0 360492  10336  56600    0    0    20    89    0   24  1  8 90  0  0
 1  0      0 360476  10344  56604    0    0     0     2    0    9  8 42 50  0  0
 1  0      0 360476  10344  56604    0    0     0     4    0    7  8 42 50  0  0
 1  0      0 360476  10344  56604    0    0     0     0    0    6  8 42 50  0  0
 1  0      0 360476  10344  56604    0    0     0     0    0    6  8 42 50  0  0
 1  0      0 360476  10344  56604    0    0     0     0    0    7  8 42 50  0  0
 1  0      0 360476  10344  56604    0    0     0     0    0    7  7 42 50  0  0
```

The `wa` column shows a wait time and represents a percentage of time while the system waited for I/O. The higher the percentage, the more time tasks waste nonproductively.

The `st` column shows what is known as *stolen time*. It represents the time that CPU was stolen from a guest by the hypervisor. This can mean several different things: CPU contention at the z/VM level, heavy z/VM paging, heavy virtual switch usage, and so on. The higher the number, the more time a guest will spend nonproductively.

sysstat package

The following tools are part of the sysstat package, which may not be installed automatically.

On RHEL 6.5, **sysstat** should be installed:

```
# rpm -qa | grep sysstat
sysstat-9.0.4-20.el6.s390
```

If it is not installed, install it with the command:

yum install sysstat.

To install sysstat on SLES 11 SP3, use zypper:

```
# zypper install sysstat
Loading repository data...
Reading installed packages...
Resolving package dependencies...

The following NEW package is going to be installed:
  sysstat

1 new package to install.
Overall download size: 174.0 KiB. After the operation, additional 662.0 KiB will be
used.
Continue? [y/n/?] (y): y
Retrieving package sysstat-8.1.5-7.45.24.s390x (1/1), 174.0 KiB (662.0 KiB unpacked)
Installing: sysstat-8.1.5-7.45.24 [done]
# rpm -qa|grep sysstat
sysstat-8.1.5-7.45.24
```

iostat command

Besides reporting overall system performance, iostat provides detailed input/output statistics for devices. Example 23-7 shows an output of the **iostat -x 5 2** command. It displays two samples with a 5-second interval.

Example 23-7 iostat command

```
Linux 2.6.32-358.el6.s390x (virtcook3.itso.ibm.com)     06/11/2013      _s390x_ (2 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           1.98    0.00    1.50    0.64    0.19   95.69

Device:         rrqm/s   wrqm/s     r/s     w/s   rsec/s   wsec/s avgrq-sz avgqu-sz   await  svctm  %util
dasda            15.31     0.16   17.96    0.50  1203.86     5.30    65.52     0.02    0.95   0.72   1.32
dasdb            21.18     0.00    3.94    0.00    33.67     0.00     8.54     0.00    0.00   0.00   0.00
dasdc            20.99     0.00    3.94    0.00    33.47     0.00     8.49     0.00    0.03   0.03   0.01
dasdd             5.79     1.22    6.46    1.40   105.35    20.93    16.07     0.01    0.69   0.26   0.21
dm-0              0.00     0.00   16.60    0.07   421.32     0.52    25.31     0.03    1.93   0.24   0.40
dm-1              0.00     0.00    6.10    2.14    55.33    17.11     8.79     0.02    2.74   0.20   0.16
dm-2              0.00     0.00    1.76    0.07    14.07     0.52     8.00     0.00    0.24   0.24   0.04
dm-3              0.00     0.00    2.73    0.41    21.80     3.30     8.00     0.00    0.69   0.17   0.05

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.00    0.00    0.10    0.00    0.00   99.90

Device:         rrqm/s   wrqm/s     r/s     w/s   rsec/s   wsec/s avgrq-sz avgqu-sz   await  svctm  %util
dasda             0.00     0.00    0.20    0.00     3.20     0.00    16.00     0.00    0.00   0.00   0.00
dasdb             0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00   0.00   0.00
dasdc             0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00   0.00   0.00
dasdd             0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00   0.00   0.00
dm-0              0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00   0.00   0.00
dm-1              0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00   0.00   0.00
dm-2              0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00   0.00   0.00
dm-3              0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00   0.00   0.00
```

The *avgqu-sz* column shows the average queue size for a given device. The larger the number, the more contention exists for a device.

The *await* column displays average wait time for a device. It includes the time that requests spent in queue and the time spent servicing requests in a storage device. The higher the number, the more time is wasted by a program waiting for I/O.

sar and sadc commands

Ad hoc data can be gathered by calling the data collector with the **sadc** command:

```
# /usr/lib64/sa/sadc -S ALL -F 5 60 /tmp/sadc.out
```

Data will be collected for 5 minutes in 5-second intervals (5x60 = 300 s = 5 minutes).

Output that is produced by sadc is a binary file. To process it and generate a text output, the **sar** command is used:

```
# sar -A -f /tmp/sadc.output > outfile.txt
```

Sar generates much detailed performance information.

If sar is configured as a service and it gathers data automatically, its data is stored in the /var/log/sa directory. Data files are of the form sa<dd> where dd is the day of the month. Text files are of the form sar<dd>. For example:

```
# cd /var/log/sa
# file *
sa09:  data
sa10:  data
sa11:  data
sa12:  data
sa13:  data
sar09: ASCII text
sar10: ASCII text
sar11: ASCII text
sar12: ASCII text
```

24

Working with disks

"*Learn from yesterday, live for today, hope for tomorrow. The important thing is not to stop questioning.*"

— Albert Einstein

This chapter has the following sections related to working with disks, both extended count key data (ECKD) DASD, and FCP/SCSI tasks that you might want to perform:

24.1 Add disk space to virtual machines

This section describes how to add additional disk space to a Linux virtual machine. The following sections provide the overall steps:

1. "Add minidisks to a virtual machine"
2. "Make the new minidisks available" on page 438

24.1.1 Add minidisks to a virtual machine

This section provides the high-level steps to add two new 3390-9s to our `LINUX3` virtual machine. The first 3390-9 is split in half to show an example of creating a new logical volume. The second 3390-9 is used to extend the existing logical volume mounted over `/var/`.

To add new minidisks to a virtual machine, perform the following steps:

1. Determine the volumes that will be added. In this example, a 3390-9 at real device address `1260` is split into two smaller minidisks, and a 3390-9 at address `1261` is added as a larger minidisk.

2. Format the volumes and attach to SYSTEM. This can be done manually, one at a time, with the CP command **CPFMTXA**, or to multiple volumes with the **CPFORMAT** EXEC, supplied in the tar file associated with this book.
3. Add minidisk statements to define minidisks. In this example, three minidisks at virtual addresses 102-104 are defined to LINUX3. The following is the updated directory entry:

```
USER LINUX3 LNX4VM 512M 1G G
   INCLUDE LNXDFLT
   OPTION APPLMON
   MDISK 0100 3390 0001 5008 JM1268 MR LNX4VM LNX4VM LNX4VM
   MDISK 0101 3390 5009 5008 JM1268 MR LNX4VM LNX4VM LNX4VM
   MDISK 0102 3390 0001 5008 JM1260 MR LNX4VM LNX4VM LNX4VM
   MDISK 0103 3390 5009 5008 JM1260 MR LNX4VM LNX4VM LNX4VM
   MDISK 0104 3390 0001 10016 JM1261 MR LNX4VM LNX4VM LNX4VM
```

4. Bring the changes to the user directory online either with **DIRECTXA** or your directory maintenance product.

Important: If you add disks to the user directory for a certain virtual machine, it is possible to attach them to a running Linux system without “bouncing” it.

For example, if you added a new minidisk at virtual address 104, you can use the following commands to link to the disk and then enable it:

```
# vmcp link '* 104 104 mr'
# chccwdev -e 104
```

5. Shut down the Linux system.
6. Log off the virtual machine.
7. Log back on to it and IPL Linux.

24.1.2 Make the new minidisks available

To make the new minidisks available, perform the following steps:

1. When your system comes back up, **start an SSH session** to it. Use the **lsdasd** command to verify that the new minidisks are not seen yet:

```
# lsdasd
Bus-ID     Status      Name      Device  Type  BlkSz  Size      Blocks
==============================================================================
0.0.0301   active      dasda     94:0    FBA   512    512MB     1048576
0.0.0300   active      dasdb     94:4    FBA   512    256MB     524288
0.0.0100   active      dasdc     94:8    ECKD  4096   3521MB    901440
0.0.0101   active      dasdd     94:12   ECKD  4096   3521MB    901440
```

2. Enable the disks with the **chccwdev -e** command:

```
# chccwdev -e 102 103 104
Setting device 0.0.0102 online
Done
Setting device 0.0.0103 online
Done
Setting device 0.0.0104 online
Done
```

3. View the available disks again with the **lsdasd** command:

```
# lsdasd
Bus-ID     Status      Name      Device  Type  BlkSz  Size      Blocks
==============================================================================
0.0.0301   active      dasda     94:0    FBA   512    512MB     1048576
0.0.0300   active      dasdb     94:4    FBA   512    256MB     524288
0.0.0100   active      dasdc     94:8    ECKD  4096   3521MB    901440
0.0.0101   active      dasdd     94:12   ECKD  4096   3521MB    901440
0.0.0102   n/f         dasde     94:16   ECKD
0.0.0103   n/f         dasdf     94:20   ECKD
0.0.0104   n/f         dasdg     94:24   ECKD
```

4. Format the disks in parallel with the **dasdfmt** command using a **for** loop and putting them in the background:

```
# for i in e f g
> do
>   dasdfmt -b 4096 -y -f /dev/dasd$i &
> done
[1] 1923
[2] 1924
[3] 1925
[root@virtcook3 ~]# Finished formatting the device.
Rereading the partition table... ok
Finished formatting the device.
Rereading the partition table... ok
Finished formatting the device.
Rereading the partition table... ok

[1]   Done                    dasdfmt -b 4096 -y -f /dev/dasd$i
[2]-  Done                    dasdfmt -b 4096 -y -f /dev/dasd$i
[3]+  Done                    dasdfmt -b 4096 -y -f /dev/dasd$i
```

5. Create one partition from each of the minidisks using a bash **for** loop and the **fdasd -a** command:

```
# for i in e f g
> do
>   fdasd -a /dev/dasd$i
> done
reading volume label ..: VOL1
reading vtoc ..........: ok
auto-creating one partition for the whole disk...
...
```

6. **If you are adding disks to RHEL 6.4,** make a backup of `/etc/dasd.conf`, then add minidisks `102`, `103` and `104` to it:

```
# cd /etc
# cp dasd.conf dasd.conf.orig
# vi dasd.conf
0.0.0301 use_diag=0 readonly=0 erplog=0 failfast=0
0.0.0300 use_diag=0 readonly=0 erplog=0 failfast=0
0.0.0101 use_diag=0 readonly=0 erplog=0 failfast=0
0.0.0100 use_diag=0 readonly=0 erplog=0 failfast=0
0.0.0102 use_diag=0 readonly=0 erplog=0 failfast=0
0.0.0103 use_diag=0 readonly=0 erplog=0 failfast=0
0.0.0104 use_diag=0 readonly=0 erplog=0 failfast=0
```

7. If you are adding disks to SLES 11 SP3, use the **dasd_configure** command to enable minidisks 102, 103, and 104:

```
# dasd_configure 0.0.0102 1
Configuring device 0.0.0102
Setting device online
# dasd_configure 0.0.0103 1
Configuring device 0.0.0103
Setting device online
# dasd_configure 0.0.0104 1
Configuring device 0.0.0104
Setting device online
```

The three new minidisks should now be formatted, partitioned, and configured to be active at boot time.

If you are creating a new logical volume, see 24.2.1, "Create a logical volume and file system" on page 440. If you are extending an existing logical volume, skip ahead to 24.3, "Extend an existing logical volume" on page 444.

24.2 Add 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 for many users to write data to. When this is the case, you can use the Logical Volume Manager (LVM) to combine multiple DASD volumes into one logical volume. This example does not create a large logical volume, but it shows all the steps necessary to do so.

The following sections describe a logical volume with additional DASD on a Linux guest. Use the following overall steps in adding a logical volume:

1. "Create a logical volume and file system"
2. "Update the file system table" on page 443

24.2.1 Create a logical volume and file system

The following overall steps are involved in creating a logical volume:

1. Create physical volumes from the two partitions
2. Create a single volume group
3. Create a single logical volume
4. Make a file system from the logical volume

Figure 24-1 on page 441 shows a block diagram of the logical volume manager.

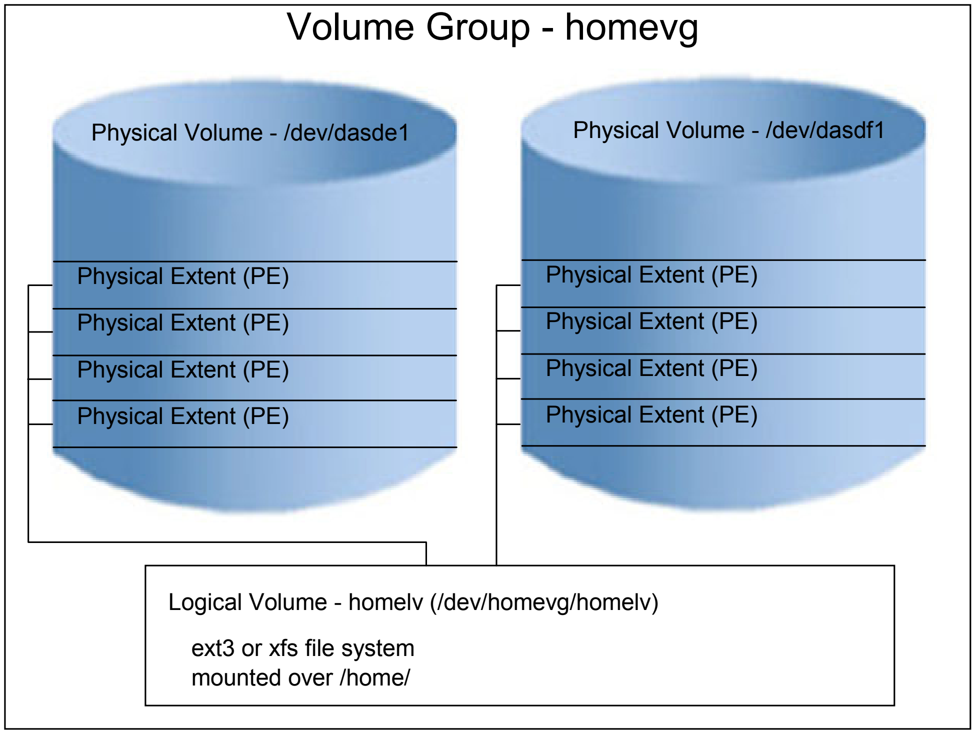


Figure 24-1 LVM block diagram

Create physical volumes from the two minidisks

To create physical volumes from the new minidisks at virtual device addresses `102` and `103`, perform the following steps:

1. The **`pvcreate`** command initializes partitions for use by LVM. Initialize the two new DASD partitions:

```
# pvcreate /dev/dasde1 /dev/dasdf1
  Physical volume "/dev/dasde1" successfully created
  Physical volume "/dev/dasdf1" successfully created
```

2. Verify that the physical volumes were created with the **`pvdisplay`** command:

```
# pvdisplay /dev/dasde1 /dev/dasdf1
  "/dev/dasde1" is a new physical volume of "3.44 GiB"
  --- NEW Physical volume ---
  PV Name               /dev/dasde1
  VG Name
  PV Size               3.44 GiB
  Allocatable           NO
  PE Size               0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               s0ugfl-hlV3-fYnf-1adW-4mOI-4HTJ-HdAOTU

  "/dev/dasdf1" is a new physical volume of "3.44 GiB"
  --- NEW Physical volume ---
  PV Name               /dev/dasdf1
  VG Name
```

```
  PV Size               3.44 GiB
  Allocatable           NO
  PE Size               0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               vO2PJY-gy4x-M9Hj-kt51-TO4J-B4n5-Ntvkje
```

Create a single volume group

The **vgcreate** command is used to create a volume group named `homevg` from the two partitions. Use the **vgdisplay homevg** command to verify that the volume group was created:

```
# vgcreate homevg /dev/dasde1 /dev/dasdf1
Volume group "homevg" successfully created
# vgdisplay homevg
  --- Volume group ---
  VG Name               homevg
  System ID
  Format                lvm2
  Metadata Areas        2
  Metadata Sequence No  1
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                0
  Open LV               0
  Max PV                0
  Cur PV                2
  Act PV                2
  VG Size               6.88 GiB
  PE Size               4.00 MiB
  Total PE              1760
  Alloc PE / Size       0 / 0
  Free  PE / Size       1760 / 6.88 GiB
  VG UUID               acSF65-56Ie-kVoY-Af6I-Hma4-VVuN-ggJEs5
```

In this example, there are 1760 free physical extents.

Create a single logical volume

In this section, you will create a single logical volume using the **lvcreate** command.

1. The **lvcreate** command is used to create a logical volume. The **-i** (a lower case I) specifies the number of stripes, in this case two, because there are two volumes in the volume group. The **-l** (a lowercase L) flag specifies the number of logical extents, 1760 in this example. The **-n homelv** specifies the name of the new logical volume. The last argument **homevg** specifies the name of the volume group from which the logical volume will be created:

```
# lvcreate -i 2 -l 1760 -n homelv homevg
  LUsing default stripesize 64.00 KiB
  Logical volume "homelv" created
```

2. Use the **lvdisplay** command to verify. The parameter is the full path of the logical volume, not just the logical volume name:

```
# lvdisplay /dev/homevg/homelv
  --- Logical volume ---
```

```
  LV Path                /dev/homevg/homelv
  LV Name                homelv
  VG Name                homevg
  LV UUID                qNcyDp-Eeqs-gfBl-XU5Z-Jt3K-QfvV-pf3Kos
  LV Write Access        read/write
  LV Creation host, time virtcook3.itso.ibm.com, 2013-06-17 15:32:39 -0400
  LV Status              available
  # open                 0
  LV Size                6.88 GiB
  Current LE             1760
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     512
  Block device           253:4
```

Make a file system from the logical volume

Create a file system from the new logical volume.

If you are on RHEL 6.4, ext4 is the recommended file system. Create an ext4 file system on the new logical volume using the **mkfs.ext4** command:

```
# mkfs.ext4 /dev/homevg/homelv
...
This filesystem will be automatically checked every 26 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
```

If you are on SLES 11 SP3, XFS is the recommended file system for data. Use the following command to make the file system.

```
# mkfs.xfs /dev/homevg/homelv
...
```

The file system created from the logical volume is now ready to be mounted.

24.2.2 Update the file system table

You could at this point mount the file system manually. However, if you add the mount to the file system table file, /etc/fstab, you can effectively test the change by using the **mount** command with only one argument. Perform the following steps:

1. Make a backup copy of the file then add the following line to it:

```
# cd /etc
# cp fstab fstab.works
```

2. Add one line to the fstab file:

```
# vi fstab
... // For RHEL 6.4:
/dev/homevg/homelv      /home                   ext4    defaults        0 0
... // For SLES 11 SP3:
/dev/homevg/homelv      /home                   xfs     defaults        0 0
...
```

3. Before mounting over /home/, you might want to check that it is empty. If a non-root user exists and a new file system is mounted over it, the contents of the directory will be *hidden*. In this example there is no data in the file system:

```
# ls -a /home
.  ..
```

4. Mount the /home/ file system with one argument. By using just one argument, you are testing the change to /etc/fstab. Use the **df -h** command to verify that it is mounted:

```
# mount /home
# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/dasdc1          1008M  184M  774M  20% /
tmpfs                 246M     0  246M   0% /dev/shm
/dev/mapper/system_vg-opt_lv
                      504M   17M  462M   4% /opt
/dev/mapper/system_vg-tmp_lv
                      504M   17M  462M   4% /tmp
/dev/mapper/system_vg-usr_lv
                      2.0G  1.3G  617M  68% /usr
/dev/mapper/system_vg-var_lv
                      504M   92M  388M  20% /var
/dev/mapper/homevg-homelv
                      6.8G  144M  6.3G   3% /home
```

5. Test a reboot to verify the new logical volume is successfully mounted over /home/:

```
# reboot
Broadcast message from root@virtcook3.itso.ibm.com
        (/dev/pts/0) at 15:51 ...

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

When the system comes back, you should see the new logical volume mounted over /home/.

24.3 Extend 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 /var/ file system is filling up on LINUX3:

```
# df -h /var/
Filesystem            Size  Used Avail Use% Mounted on
/dev/mapper/system_vg-var_lv
                      504M  392M   88M  82% /var
```

A 3390-9 was added as minidisk 106 in section 24.1, "Add disk space to virtual machines" on page 437.

> **Attention:** It is possible to attach minidisks to a running Linux system without rebooting it. For example, if you added a new minidisk at virtual address 106, from a root SSH session, use the command: **vmcp link * 106 106 mr** to link to the minidisk, then **chccwdev -e 106** to enable it.

To extend the logical volume using this disk, perform the following steps:

1. Use the **vgdisplay** command to see the free space in the volume group `system_vg`:

```
# vgdisplay system_vg
  --- Volume group ---
  VG Name               system_vg
  System ID
  Format                lvm2
  Metadata Areas        2
  Metadata Sequence No  6
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                5
  Open LV               4
  Max PV                0
  Cur PV                2
  Act PV                2
  VG Size               5.88 GiB
  PE Size               4.00 MiB
  Total PE              1504
  Alloc PE / Size       1504 / 5.88 GiB
  Free  PE / Size       0 / 0
  VG UUID               4i89gF-b0xm-dkHo-blWP-3Kca-0xCI-V6TAXk
```

   This shows that there are no free extents in the volume group

2. Use the **lsdasd** command to show the enabled disks:

```
# lsdasd
Bus-ID     Status      Name      Device  Type  BlkSz  Size      Blocks
==============================================================================
0.0.0100   active      dasda     94:0    ECKD  4096   3521MB    901440
0.0.0301   active      dasdb     94:4    FBA   512    512MB     1048576
0.0.0300   active      dasdc     94:8    FBA   512    256MB     524288
0.0.0101   active      dasdd     94:12   ECKD  4096   3521MB    901440
0.0.0102   active      dasde     94:16   ECKD  4096   3521MB    901440
0.0.0103   active      dasdf     94:20   ECKD  4096   3521MB    901440
0.0.0104   active      dasdg     94:24   ECKD  4096   7042MB    1802880
```

   This shows that minidisk `104` is at `/dev/dasdg`.

3. Make minidisk `104` a physical volume with the **pvcreate** command:

```
# pvcreate /dev/dasdg1
  Physical volume "/dev/dasdg1" successfully created
```

4. Use the **vgextend** command to add the minidisk to the volume group:

```
# vgextend system_vg /dev/dasdg1
  Volume group "system_vg" successfully extended
```

5. Use the **vgdisplay** command again to show the free extents in the volume group:

```
# vgdisplay system_vg
  --- Volume group ---
  VG Name               system_vg
  System ID
  Format                lvm2
  Metadata Areas        3
  Metadata Sequence No  7
```

```
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                5
  Open LV               4
  Max PV                0
  Cur PV                3
  Act PV                3
  VG Size               12.75 GiB
  PE Size               4.00 MiB
  Total PE              3264
  Alloc PE / Size       1504 / 5.88 GiB
  Free  PE / Size       1760 / 6.88 GiB
  VG UUID               4i89gF-b0xm-dkHo-blWP-3Kca-0xCI-V6TAXk
```

This shows that there are now 1760 free extents in the volume group.

6. Use the **mount** command to determine the name of the logical volume mounted over /var/:

```
# mount | grep "\/var "
/dev/mapper/system_vg-var_lv on /var type ext4 (rw)
```

In this example, it is /dev/mapper/system_vg-var_lv/.

7. Use the **lvextend** command to extend the volume group with all of the new extents:

```
# lvextend -l +1760 /dev/mapper/system_vg-var_lv
  Extending logical volume var_lv to 7.38 GiB
  Logical volume var_lv successfully resized
```

8. Use the **resize2fs** command to increase the size of the EXT4file system while it is still mounted:

```
# resize2fs /dev/mapper/system_vg-var_lv
resize2fs 1.41.12 (17-May-2010)
Filesystem at /dev/mapper/system_vg-var_lv is mounted on /var; on-line resizing
required
old desc_blocks = 1, new_desc_blocks = 1
Performing an on-line resize of /dev/mapper/system_vg-var_lv to 1933312 (4k)
blocks.
The filesystem on /dev/mapper/system_vg-var_lv is now 1933312 blocks long.
```

9. Use the **xfs_growfs** command to increase the size of the XFS file system while it is still mounted:

```
# xfs_growfs /dev/mapper/system_vb-var_lv
```

10. Use the **df** command to show the file system size before and after you extend it, as the following example shows:

```
# df -h /var
Filesystem            Size  Used Avail Use% Mounted on
/dev/mapper/system_vg-var_lv
                      7.3G  393M  6.6G   6% /var
```

This shows that the /var/ file system now has 6.6 GB of free space.

24.4 Moving a physical volume

Besides file systems getting larger, you might have a need to move data off one or more volumes onto another or target set of volumes. If your data is in LVM, the **pvmove** and **vgreduce** commands were designed for this, and can be used with the file system online.

In this example, two physical volumes exist, `/dev/dasde1` and `/dev/dasdf1`. Data is populated on the first volume, and later moved to the second. This is done while the file system is online.

To complete this test, perform the following steps:

- Create a volume group from the first logical volume. In this example, it is named `homelv`:

```
# vgcreate homevg /dev/dasde1
  Volume group "homevg" successfully created
```

- Observe the number of physical extents:

```
# vgdisplay homevg | grep "Total PE"
  Total PE              1760
```

- Create a logical volume from the volume group. In this example, it is named `homelv` and all physical extents are used:

```
# lvcreate -l 1760 -n homelv homevg
  Logical volume "homelv" created
```

- Create a file system from the logical volume. In this example, it is of type ext4:

```
# mkfs.ext4 /dev/homevg/homelv
```

- Add the new file system to the file system table and mount it:

```
# vi /etc/fstab
...
# grep home /etc/fstab
/dev/homevg/homelv      /home                   ext4    defaults        0 0
# mount /home
```

- Create a sizable file on it with the **dd** command and show file system usage:

```
# dd if=/dev/zero of=/home/bigfile bs=1M count=500
500+0 records in
500+0 records out
524288000 bytes (524 MB) copied, 3.0718 s, 171 MB/s
# df -h | grep home
/dev/mapper/homevg-homelv
                      6.8G  644M  5.8G  10% /home
```

- Show the volume group usage with the **vgdisplay** command:

```
# vgdisplay homevg
  --- Volume group ---
  VG Name               homevg
  VG Size               6.88 GiB
  PE Size               4.00 MiB
  Total PE              1760
  Alloc PE / Size       1760 / 6.88 GiB
  Free  PE / Size       0 / 0
  VG UUID               YIQgoN-865f-3Vbf-tjH1-eXhO-Aa6W-PcxHri
```

  This shows that all physical extents in the volume group are used.

- Add a second physical volume that will be the target of the data move, to the volume group:

```
# vgextend homevg /dev/dasdf1
```

```
  Volume group "homevg" successfully extended
```

- Show the volume group usage again:

```
# vgdisplay homevg
  --- Volume group ---
  VG Name               homevg
  ...
  VG Size               13.75 GiB
  PE Size               4.00 MiB
  Total PE              3520
  Alloc PE / Size       1760 / 6.88 GiB
  Free  PE / Size       1760 / 6.88 GiB
  VG UUID               YIQgoN-865f-3Vbf-tjH1-eXhO-Aa6W-PcxHri
```

This shows that the volume group doubled in size and there are now an equal number of free extents.

- Move the data off the source physical volume with the **pvmove** command. The target need not be specified:

```
# pvmove /dev/dasde1
  /dev/dasde1: Moved: 0.0%
  /dev/dasde1: Moved: 8.0%
  /dev/dasde1: Moved: 18.9%
  /dev/dasde1: Moved: 34.2%
  /dev/dasde1: Moved: 49.1%
  /dev/dasde1: Moved: 63.2%
  /dev/dasde1: Moved: 77.6%
  /dev/dasde1: Moved: 92.7%
  /dev/dasde1: Moved: 100.0%
```

- Show the volume group usage again:

```
# vgdisplay homevg
  --- Volume group ---
  VG Name               homevg
  ...
  VG Size               13.75 GiB
  PE Size               4.00 MiB
  Total PE              3520
  Alloc PE / Size       1760 / 6.88 GiB
  Free  PE / Size       1760 / 6.88 GiB
  VG UUID               YIQgoN-865f-3Vbf-tjH1-eXhO-Aa6W-PcxHri
```

These free and used extents are the same, however, the data has been moved.

- Show the free and used extents on the source and target physical volumes with the **pvdisplay** command:

```
# pvdisplay /dev/dasde1 /dev/dasdf1
  --- Physical volume ---
  PV Name               /dev/dasde1
  VG Name               homevg
  PV Size               6.88 GiB / not usable 2.41 MiB
  Allocatable           yes
  PE Size               4.00 MiB
  Total PE              1760
  Free PE               1760
  Allocated PE          0
  PV UUID               Jo2fa3-5cc0-y2Xs-e0DQ-wQXc-i3er-MPcckW

  --- Physical volume ---
  PV Name               /dev/dasdf1
  VG Name               homevg
```

```
  PV Size               6.88 GiB / not usable 2.41 MiB
  Allocatable           yes (but full)
  PE Size               4.00 MiB
  Total PE              1760
  Free PE               0
  Allocated PE          1760
  PV UUID               hme2qP-6ytn-Drg8-Wba4-rTU1-q1sV-pVZ03g
```

- Remove the source physical volume:

```
# vgreduce homevg /dev/dasde1
  Removed "/dev/dasde1" from volume group "homevg"
```

The source volume should now be ready for reassignment, or retirement.

This section has shown how to move data from one physical volume to another without taking the file system offline.

24.5 Add SCSI/FCP disks

This book has only described ECKD disks, also known as DASD. In addition, z/VM and Linux support SCSI/FCP disks.

The Fibre Channel (FC) standard was developed by the National Committee of Information Technology Standards (NCITS). The System z FCP I/O architecture conforms to these standards. System z FCP support enables z/VM and Linux running on System z to access industry-standard SCSI devices. For disk applications, these FCP storage devices utilize Fixed Block (512-byte) sectors rather than ECKD format. A new channel-path identifier (CHPID) type has been defined called `FCP`. The `FCP` CHPID type is supported on the FICON and FICON Express features of all System z processors.

This is only a brief introduction to SCSI/FCP disks and multipathing. For more complete documentation, see the Redbooks publication *Fibre Channel Protocol for Linux and z/VM on IBM System z* on the web at the following site:

http://www.redbooks.ibm.com/abstracts/sg247266.html?Open

In addition, see the Redbooks publication *Introducing N_Port Identifier Virtualization for IBM System z9®*, on the web at the following site:

http://www.redbooks.ibm.com/abstracts/redp4125.html?Open

When using Fibre Channel Protocol (FCP), it is a common best practice to keep redundancy for disk access on several levels. There should be at least two FCP adapters used on the server side, each connected to a separate SAN fabric. The storage device should be also connected by at least two adapters to those fabrics, as shown in Figure 24-2 on page 450. If several operating systems are sharing the same physical FCP adapters on a server, it is necessary to use NPIV-enabled switches. This way, each operating system will have its own worldwide port names (WWPNs), which allow to use zoning in SAN and logical unit number (LUN) masking in the storage device to set up proper access from the operating system to appropriate LUNs.

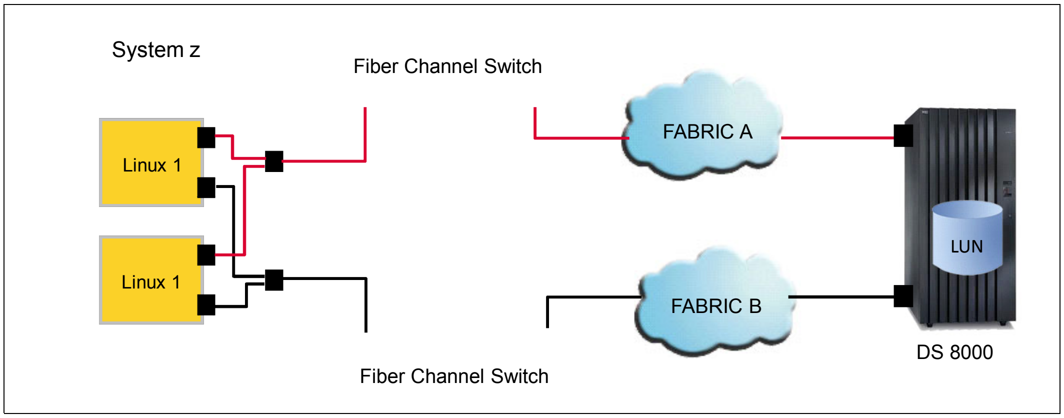


Figure 24-2 Connection overview between storage and guests

In general, there are two reasons for establishing multiple paths to a device:

- High availability: Providing several physical paths to a device offers high availability. If one path fails, other paths to the device are still available and processing can continue uninterrupted.
- High performance: Using multiple paths simultaneously to read from or to write to a device can significantly increase I/O performance.

Multipathing for ECKD volumes is handled in the hardware and the operating system does not need to be aware of the multipathing. However, redundant paths to SCSI LUNs appear to the operating system as two or more disks, each pointing to the same LUN. From the operating system point of view, they appear as different disks. For this reason, it is necessary to use a software layer, which hides multiple instances of the same LUN under one virtual device name. In Linux, multipath daemon is used for this.

Figure 24-3 on page 451 shows our FCP environment used in this chapter.

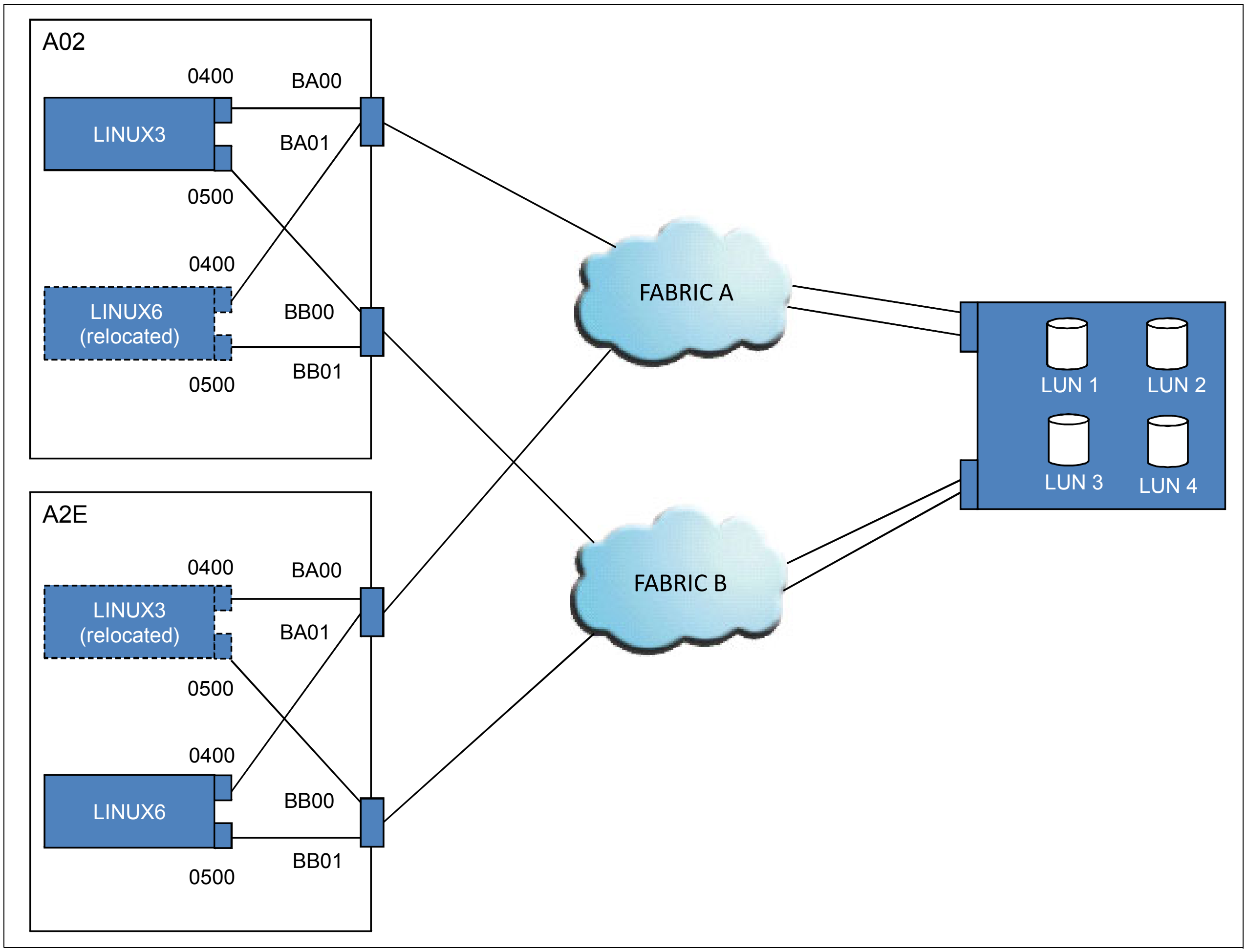


Figure 24-3 Our FCP environment

From a Linux point of view, there are two FCP host adapters with addresses `400` and `500` and unique WWPNs. It is z/VM, which presents dedicated real devices `BAxx` and `BBxx` to the Linux guest as virtual devices `400` and `500`. It is the NPIV protocol, which allows many different WWPNs on the same physical adapter. Both adapters have access to two different fabrics, Fabric A, and Fabric B. These are then connected to the storage system to two different controllers, which in turn have access to all of the needed LUNs.

Table 24-1 summarizes all WWPNs related to our setup.

Table 24-1 WWPN list

Adapter	WWPN	Usage
LPAR A02 device `BA00`	`c05076ecea800c20`	`LINUX3` device `400` on LPAR A02
LPAR A02 device `BA01`	`c05076ecea800c24`	`LINUX6` device `400` on LPAR A02
LPAR A02 device `BB00`	`c05076ecea800d30`	`LINUX3` device `500` on LPAR A02
LPAR A02 device `BB01`	`c05076ecea800d34`	`LINUX6` device `500` on LPAR A02
LPAR A2E device `BA00`	`c05076e5f9002350`	`LINUX3` device `400` on LPAR A2E
LPAR A02 device `BA01`	`c05076e5f9002354`	`LINUX6` device `400` on LPAR A2E

Adapter	WWPN	Usage
LPAR A02 device BB00	c05076e5f90024e8	LINUX3 device 500 on LPAR A2E
LPAR A02 device BB01	c05076e5f90024ec	LINUX6 device 500 on LPAR A2E
DS8300 adapter A	500507630500c74c	Storage access
DS8300 adapter B	500507630508c74c	Storage access

Table 24-2 shows a list of LUNs used in this section.

Table 24-2 LUN list

LUN	Usage
4010401100000000	LUN 1 for LINUX3
4011401100000000	LUN 2 for LINUX3
4010401200000000	LUN 3 for LINUX6
4011401200000000	LUN 4 for LINUX6

Linux guests have virtual FCP adapters with addresses 0400 and 0500 regardless of the real device numbers. This way all Linux guests look the same and they are easier to manage.

To be able to relocate a running Linux guest with FCP devices from one LPAR to another, FCP device numbers on each LPAR have to be the same and EQID has to be set up.

To create an EQID dynamically use the **SET RDEVICE** command. Execute it on each single system image (SSI) LPAR where Linux should be able to relocate. Follow these steps to accomplish this:

1. Start a 3270 session as MAINT.
2. Vary devices offline:

```
==> vary off ba00-ba01 bb00-bb01
BA00 varied offline
BA01 varied offline
BB00 varied offline
BB01 varied offline
4 device(s) specified; 4 device(s) successfully varied offline
```

3. Use **SET RDEVICE** to create EQIDs dynamically:

```
===> set rdev ba00 eqid fcpid00 type fcp
HCPZRP6722I Characteristics of device BA00 were set as requested.
1 RDEV(s) specified; 1 RDEV(s) changed; 0 RDEV(s) created
===> set rdev bb00 eqid fcpid00 type fcp
HCPZRP6722I Characteristics of device BB00 were set as requested.
1 RDEV(s) specified; 1 RDEV(s) changed; 0 RDEV(s) created
===> set rdev ba01 eqid fcpid01 type fcp
HCPZRP6722I Characteristics of device BA01 were set as requested.
1 RDEV(s) specified; 1 RDEV(s) changed; 0 RDEV(s) created
===> set rdev bb01 eqid fcpid01 type fcp
HCPZRP6722I Characteristics of device BB01 were set as requested.
1 RDEV(s) specified; 1 RDEV(s) changed; 0 RDEV(s) created
```

4. Check the result with the **QUERY EQID** command:

```
===> q eqid fcpid00
Devices for FCPID00:
BA00 BB00
===> q eqid fcpid01
Devices for FCPID01:
BA01 BB01
```

5. Vary the devices online with the **VARY ON** command:

```
==> vary on ba00-ba01 bb00-bb01
BA00 varied online
BA01 varied online
BB00 varied online
BB01 varied online
4 device(s) specified; 4 device(s) successfully varied online
```

To make the EQIDs permanent, perform the following steps:

6. Edit the `SYSTEM CONFIG` file and add **RDEV** statements:

```
==> link pmaint cf0 cf0 mr
==> acc cf0 f
==> x system config f
/* Add EQID statements for OSA addresses, unique MAC IDs and FCP*/
ZVM63A: begin
  rdev 2100-210f eqid osaset1 type osa
  rdev 2120-212f eqid osaset1 type osa
  vmlan macprefix 02000b
  vmlan limit transient 0
  define vswitch vsw1 rdev 2103 2123 ethernet
  define vswitch vsw2 ethernet
  RDEV BA00 EQID FCPID00 TYPE FCP
  RDEV BA01 EQID FCPID01 TYPE FCP
  RDEV BB00 EQID FCPID00 TYPE FCP
  RDEV BB01 EQID FCPID01 TYPE FCP
ZVM63A: end
ZVM63B: begin
  rdev 2040-204f eqid osaset1 type osa
  rdev 2060-206f eqid osaset1 type osa
  vmlan macprefix 02000c
  vmlan limit transient 0
  define vswitch vsw1 rdev 2043 2063 ethernet
  define vswitch vsw2 ethernet
  RDEV BA00 EQID FCPID00 TYPE FCP
  RDEV BA01 EQID FCPID01 TYPE FCP
  RDEV BB00 EQID FCPID00 TYPE FCP
  RDEV BB01 EQID FCPID01 TYPE FCP
ZVM63B: end
```

7. Check the syntax of the change with the **CPSYNTAX** command on the `MAINT 193` disk:

```
==> acc 193 g
==> cpsyntax system config f (lpar a02
CONFIGURATION FILE PROCESSING COMPLETE -- NO ERRORS ENCOUNTERED.
==> cpsyntax system config f (lpar a2e
CONFIGURATION FILE PROCESSING COMPLETE -- NO ERRORS ENCOUNTERED.
```

When z/VM IPLs, the EQIDs will now be created.

24.5.1 Define FCP to Linux virtual machines

You can determine the status of FCP devices in your LPAR with the z/VM **QUERY FCP** and **QUERY FCP FREE** commands. The following is an example from a MAINT 3270 session:

```
==> q fcp free
FCP  BA00 FREE    , FCP  BA01 FREE    , FCP  BA02 FREE    , FCP  BA03 FREE
FCP  BA04 FREE    , FCP  BA05 FREE    , FCP  BA06 FREE    , FCP  BA07 FREE
...
FCP  BB00 FREE    , FCP  BB01 FREE    , FCP  BB02 FREE    , FCP  BB03 FREE
FCP  BB04 FREE    , FCP  BB05 FREE    , FCP  BB06 FREE    , FCP  BB07 FREE
...
```

The output shows that the LPAR has many free FCP devices. But only FCP devices which WWPN were defined in zoning and LUN masking will be able to access SCSI LUNs. In our case, we are using ***BA00*** and ***BB00*** for LINUX3 running RHEL and devices ***BA01*** and ***BB01*** for LINUX6 running SLES.

To attach FCP devices dynamically, perform the following steps:

1. On the MAINT 3270 session and attach an FCP device to Linux guest with the **ATTACH** command:

   ```
   For LINUX3:
   ==> att ba00 linux3 as 400
   FCP BA00 ATTACHED TO LINUX3 0400
   ==> att bb00 linux3 as 500
   FCP BB00 ATTACHED TO LINUX3 0500

   For LINUX6:
   ==> att ba01 linux6 as 400
   FCP BA01 ATTACHED TO LINUX6 0400
   ==> att bb01 linux6 as 500
   FCP BB01 ATTACHED TO LINUX6 0500
   ```

2. **QUERY FCP** command shows attached FCP devices together with their virtual WWPNs:

   ```
   On ZVM63A:
   ==> q fcp
   FCP  BA00 ATTACHED TO LINUX3   0400 CHPID 4A
        WWPN C05076ECEA800C20
   FCP  BB00 ATTACHED TO LINUX3   0500 CHPID 4B
        WWPN C05076ECEA800D30
   On ZVM63B:
   ==> q fcp
   FCP  BA01 ATTACHED TO LINUX6   0400 CHPID 7A
        WWPN C05076E5F9002354
   FCP  BB01 ATTACHED TO LINUX6   0500 CHPID 7B
        WWPN C05076E5F90024EC
   ```

Make the changes persistent in one of the following ways:

- If using Dirmaint, perform each of the following commands:

  ```
  ==> dirm for linux3 dedicate 400 ba00
  ==> dirm for linux3 dedicate 500 bb00
  ==> dirm for linux6 dedicate 400 ba01
  ==> dirm for linux6 dedicate 500 bb01
  ```

- If not using Dirmaint, add a **DEDICATE** statement to virtualize BA00 and BB00 (which is the FCP device) as virtual devices 400 and 500:

```
USER LINUX3 LNX4VM 512M 1G G
 INCLUDE LNXDFLT
 ...
 DEDICATE 400 BA00
 DEDICATE 500 BB00
...
USER LINUX6 LNX4VM 512M 1G G
 INCLUDE LNXDFLT
 ...
 DEDICATE 400 BA01
 DEDICATE 500 BB01
...
```

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

24.5.2 FCP/SCSI disks in RHEL

To enable the FCP/SCSI disks in RHEL 6.3, perform the following steps:

1. Start an SSH session as root to LINUX3.
2. Verify that the zfcp module is loaded with the following command:

```
# lsmod | grep zfcp
zfcp                  148126  0 [permanent]
scsi_transport_fc      73035  1 zfcp
scsi_mod              297395  3 zfcp,scsi_transport_fc,scsi_tgt
qdio                   71293  3 zfcp,qeth_l2,qeth
```

3. Change directory to /sys/bus/ccw/drivers/ and list the contents:

```
# cd /sys/bus/ccw/drivers
# ls -F
3215/  3270/  dasd-eckd/  dasd-fba/  qeth/  vmur/  zfcp/
```

Note that there is a directory named zfcp/.

4. Change into that directory and list the contents, note the symbolic links to 0.0.ba00 and 0.0.bb00:

```
# cd zfcp
# ls -F
lrwxrwxrwx. 1 root root    0 Jun 19 15:29 0.0.0400 ->
../../../../devices/css0/0.0.0000/0.0.0400
lrwxrwxrwx. 1 root root    0 Jun 19 15:29 0.0.0500 ->
../../../../devices/css0/0.0.0001/0.0.0500
--w-------. 1 root root 4096 Jun 20 08:26 bind
lrwxrwxrwx. 1 root root    0 Jun 20 08:26 module -> ../../../../module/zfcp
--w-------. 1 root root 4096 Jun 19 15:28 uevent
--w-------. 1 root root 4096 Jun 20 08:26 unbind
```

5. Type the contents of the online file for each symbolic link:

```
# cd /sys/bus/ccw/drivers/zfcp/0.0.0400/
# cat online
0
# cd /sys/bus/ccw/drivers/zfcp/0.0.0500/
# cat online
```

```
0
```

A value of `0` shows that the devices are offline.

6. Echo a `1` into the file and it will be put online (you could also use the **chccwdev -e** command):

```
# cd /sys/bus/ccw/drivers/zfcp/0.0.0400/
# echo 1 > online
# cat online
1
# cd /sys/bus/ccw/drivers/zfcp/0.0.0500/
# echo 1 > online
# cat online
1
```

7. List the contents of the directory again. You should see that many entries were added after the device was put online. The four entries in bold are the WWPNs available from this FCP device:

```
# cd /sys/bus/ccw/drivers/zfcp/0.0.0400/
# ls -F
0x500507630500c74c  cutype            host1        peer_d_id    power
0x500507630508c74c  devtype           in_recovery  peer_wwnn    status
availability        driver            lic_version  peer_wwpn    subsystem
card_version        failed            modalias     port_remove  uevent
cmb_enable          hardware_version  online       port_rescan

# cd /sys/bus/ccw/drivers/zfcp/0.0.0500/
# ls -F
0x500507630500c74c  cutype            host0        peer_d_id    power
0x500507630508c74c  devtype           in_recovery  peer_wwnn    status
availability        driver            lic_version  peer_wwpn    subsystem
card_version        failed            modalias     port_remove  uevent
cmb_enable          hardware_version  online       port_rescan
```

8. The **lsluns** command shows all of the available LUNs from a single WWPN. In the following example, the first WWPN is used:

```
# lsluns
Scanning for LUNs on adapter 0.0.0400
        at port 0x500507630500c74c:
                0x4010401100000000
                0x4011401100000000
        at port 0x500507630508c74c:
                0x4010401100000000
                0x4011401100000000
Scanning for LUNs on adapter 0.0.0500
        at port 0x500507630500c74c:
                0x4010401100000000
                0x4011401100000000
        at port 0x500507630508c74c:
                0x4010401100000000
                0x4011401100000000
```

The output shows two LUNs (`0x4010401100000000` and `0x4011401100000000`), each accessible via four paths.

9. Bring a LUN online. In this example, the next free LUN is `0x4010401100000000`. Change directory into the first WWPN and list the contents:

```
# cd /sys/bus/ccw/drivers/zfcp/0.0.0400/
# cd 0x500507630500c74c
# ls
access_denied  in_recovery  status  unit_add
failed         power        uevent  unit_remove

# cd /sys/bus/ccw/drivers/zfcp/0.0.0400/
# cd 0x500507630508c74c
# ls
access_denied  in_recovery  status  unit_add
failed         power        uevent  unit_remove

# cd /sys/bus/ccw/drivers/zfcp/0.0.0500/
# cd 0x500507630500c74c
# ls
access_denied  in_recovery  status  unit_add
failed         power        uevent  unit_remove

# cd /sys/bus/ccw/drivers/zfcp/0.0.0500/
# cd 0x500507630508c74c
# ls
access_denied  in_recovery  status  unit_add
failed         power        uevent  unit_remove
```

The output shows that there is no active LUN under this WWPN.

10. Bring the LUN online by echoing the value into the file `unit_add` and list the contents of the directory:

```
# cd /sys/bus/ccw/drivers/zfcp/0.0.0400/
# cd 0x500507630500c74c
# echo  0x0x4010400000000000 > unit_add
# echo  0x0x4011400000000000 > unit_add
# ls -F
0x4010401100000000/  access_denied  in_recovery  status  unit_add
0x4011401100000000/  failed         power/       uevent  unit_remove

# cd /sys/bus/ccw/drivers/zfcp/0.0.0400
# cd 0x500507630508c74c
# echo  0x0x4010400000000000 > unit_add
# echo  0x0x4011400000000000 > unit_add
# ls -F
0x4010401100000000/  access_denied  in_recovery  status  unit_add
0x4011401100000000/  failed         power/       uevent  unit_remove

# cd /sys/bus/ccw/drivers/zfcp/0.0.0500/
# cd 0x500507630500c74c
# echo  0x0x4010400000000000 > unit_add
# echo  0x0x4011400000000000 > unit_add
# ls -F
0x4010401100000000/  access_denied  in_recovery  status  unit_add
0x4011401100000000/  failed         power/       uevent  unit_remove

# cd /sys/bus/ccw/drivers/zfcp/0.0.0500/
# cd 0x500507630508c74c
```

```
# echo  0x0x4010400000000000 > unit_add
# echo  0x0x4011400000000000 > unit_add
0x4010401100000000/  access_denied  in_recovery  status  unit_add
0x4011401100000000/  failed         power/       uevent  unit_remove
```

11. A new directory with the LUN value is created:

```
# lszfcp -D
0.0.0400/0x500507630500c74c/0x4010401100000000 0:0:0:1074872336
0.0.0400/0x500507630500c74c/0x4011401100000000 0:0:0:1074872337
0.0.0400/0x500507630508c74c/0x4010401100000000 0:0:1:1074872336
0.0.0400/0x500507630508c74c/0x4011401100000000 0:0:1:1074872337
0.0.0500/0x500507630508c74c/0x4010401100000000 1:0:0:1074872336
0.0.0500/0x500507630508c74c/0x4011401100000000 1:0:0:1074872337
0.0.0500/0x500507630500c74c/0x4010401100000000 1:0:1:1074872336
0.0.0500/0x500507630500c74c/0x4011401100000000 1:0:1:1074872337
```

12. To make the changes persistent, you should create the proper configuration in the file `/etc/zfcp.conf`. In this example, we used the output of **lszfcp** command line, filtered by **awk** and **sed** to create the proper format for the configuration file:

```
# lszfcp -D | awk '{ print $1 }' | sed -e 's/\// /g' >/etc/zfcp.conf
# cat /etc/zfcp.conf
0.0.0400 0x500507630500c74c 0x4010401100000000
0.0.0400 0x500507630500c74c 0x4011401100000000
0.0.0400 0x500507630508c74c 0x4010401100000000
0.0.0400 0x500507630508c74c 0x4011401100000000
0.0.0500 0x500507630508c74c 0x4010401100000000
0.0.0500 0x500507630508c74c 0x4011401100000000
0.0.0500 0x500507630500c74c 0x4010401100000000
0.0.0500 0x500507630500c74c 0x4011401100000000

# cat /proc/scsi/scsi
Attached devices:
Host: scsi0 Channel: 00 Id: 00 Lun: 1074872336
  Vendor: IBM      Model: 2107900          Rev: .120
  Type:   Direct-Access                    ANSI  SCSI revision: 05
Host: scsi0 Channel: 00 Id: 00 Lun: 1074872337
  Vendor: IBM      Model: 2107900          Rev: .120
  Type:   Direct-Access                    ANSI  SCSI revision: 05
Host: scsi0 Channel: 00 Id: 01 Lun: 1074872336
  Vendor: IBM      Model: 2107900          Rev: .120
  Type:   Direct-Access                    ANSI  SCSI revision: 05
Host: scsi0 Channel: 00 Id: 01 Lun: 1074872337
  Vendor: IBM      Model: 2107900          Rev: .120
  Type:   Direct-Access                    ANSI  SCSI revision: 05
Host: scsi1 Channel: 00 Id: 01 Lun: 1074872336
  Vendor: IBM      Model: 2107900          Rev: .120
  Type:   Direct-Access                    ANSI  SCSI revision: 05
Host: scsi1 Channel: 00 Id: 01 Lun: 1074872337
  Vendor: IBM      Model: 2107900          Rev: .120
  Type:   Direct-Access                    ANSI  SCSI revision: 05
Host: scsi1 Channel: 00 Id: 00 Lun: 1074872336
  Vendor: IBM      Model: 2107900          Rev: .120
  Type:   Direct-Access                    ANSI  SCSI revision: 05
Host: scsi1 Channel: 00 Id: 00 Lun: 1074872337
  Vendor: IBM      Model: 2107900          Rev: .120
  Type:   Direct-Access                    ANSI  SCSI revision: 05
```

Configure multipath

Although there are only two LUNs available for this Linux, the system can see eight disks `/dev/sda - /dev/sdh`. For better availability and management of multiple paths to the same LUN, set up multipath. To set up multipathing, perform the following steps:

1. Install the `device-mapper-multipath` RPM:

```
# yum -y install device-mapper-multipath
Installed:
   device-mapper-multipath.s390x 0:0.4.9-64.el6
...
```

2. Copy the file `/usr/share/doc/device-mapper-multipath-0.4.9/multipath.conf` to `/etc/multipath.conf` and edit it:

```
# cp /usr/share/doc/device-mapper-multipath-0.4.9/multipath.conf
/etc/multipath.conf
# cd /etc
# vi multipath.conf
defaults {
        user_friendly_names yes
}
# Blacklists devices that should not be part of the multipath
blacklist {
        devnode "^(ram|raw|loop|fd|md|dm-|sr|scd|st)[0-9]*"
        devnode "^hd[a-z]"
        devnode "^dasd[a-z][0-9]*"
}
```

   Adding the line to the blacklist prevents minidisks from being considered for multipathing.

3. Turn the multipath service on for this session and across reboots:

```
# service multipathd start
Starting multipathd daemon: [  OK  ]
# chkconfig multipathd on

# multipath -ll
mpathc (36005076305ffc74c0000000000001111) dm-8 IBM,2107900
size=20G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=1 status=active
  |- 1:0:1:1074872337 sdb 8:16  active ready running
  |- 1:0:0:1074872337 sdd 8:48  active ready running
  |- 0:0:0:1074872337 sdf 8:80  active ready running
  `- 0:0:1:1074872337 sdh 8:112 active ready running
mpatha (36005076305ffc74c0000000000001011) dm-6 IBM,2107900
size=20G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=1 status=active
  |- 1:0:1:1074872336 sda 8:0   active ready running
  |- 1:0:0:1074872336 sdc 8:32  active ready running
  |- 0:0:0:1074872336 sde 8:64  active ready running
  `- 0:0:1:1074872336 sdg 8:96  active ready running
```

4. Create an alias for each path to easily identify the LUNs by editing the `multipath.conf` file and add the following lines:

```
multipaths {
        multipath {
                wwid                    36005076305ffc74c0000000000001111
                alias                   lun1
```

```
        }
        multipath {
                wwid                    36005076305ffc74c0000000000001011
                alias                   lun2
        }
}
```

5. Check the modification by restarting the multipathd service and querying multipath for information about the paths:

```
# service multipathd restart
Stopping multipathd daemon: [  OK  ]
Starting multipathd daemon: [  OK  ]

# multipath -ll
lun2 (36005076305ffc74c0000000000001011) dm-6 IBM,2107900
size=20G features='0' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=1 status=active
  |- 1:0:1:1074872336 sda 8:0   active ready running
  |- 1:0:0:1074872336 sdc 8:32  active ready running
  |- 0:0:0:1074872336 sde 8:64  active ready running
  `- 0:0:1:1074872336 sdg 8:96  active ready running
lun1 (36005076305ffc74c0000000000001111) dm-8 IBM,2107900
size=20G features='0' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=1 status=active
  |- 1:0:1:1074872337 sdb 8:16  active ready running
  |- 1:0:0:1074872337 sdd 8:48  active ready running
  |- 0:0:0:1074872337 sdf 8:80  active ready running
  `- 0:0:1:1074872337 sdh 8:112 active ready running
```

6. Verify that the new `lun1` friendly name has been added:

```
# ls /dev/mapper
control  lun2              system_vg-tmp_lv  system_vg-var_lv
lun1     system_vg-opt_lv  system_vg-usr_lv
```

7. Create a partition to the multipath device with the **fdisk** command:

```
# fdisk /dev/mapper/lun1
...
Command (m for help): p
Disk /dev/mapper/lun1: 21.5 GB, 21474836480 bytes
255 heads, 63 sectors/track, 2610 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x8e81b172

            Device Boot      Start         End      Blocks   Id  System

Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-2610, default 1):
Using default value 1
Last cylinder, +cylinders or +size{K,M,G} (1-2610, default 2610):
```

```
Using default value 2610

Command (m for help): w
The partition table has been altered!

# fdisk /dev/mapper/lun2
...
Command (m for help): p

Disk /dev/mapper/lun2: 21.5 GB, 21474836480 bytes
255 heads, 63 sectors/track, 2610 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x8a4560a0

            Device Boot      Start         End      Blocks   Id  System

Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-2610, default 1):
Using default value 1
Last cylinder, +cylinders or +size{K,M,G} (1-2610, default 2610):
Using default value 2610

Command (m for help): w
The partition table has been altered!
```

8. Reread partition tables and update the multipath partition mapping:

```
# partprobe
# kpartx -l -v /dev/mapper/lun1
add map lun1p1 (253:6): 0 41929587 linear /dev/mapper/lun1 63
# kpartx -l -v /dev/mapper/lun2
add map lun2p1 (253:7): 0 41929587 linear /dev/mapper/lun2 63
# ls /dev/mapper
control  lun1p1  lun2p1            system_vg-tmp_lv  system_vg-var_lv
lun1     lun2    system_vg-opt_lv  system_vg-usr_lv
```

9. Format the multipathed partitions as ext4:

```
# mkfs.ext4 /dev/mapper/lun1p1
# mkfs.ext4 /dev/mapper/lun2p1
```

10. Mount the multipathed LUN with the new name and see that the test file exists:

```
# mkdir -p /mnt/lun1
# mkdir -p /mnt/lun2
# mount /dev/mapper/lun1p1 /mnt/lun1
# mount /dev/mapper/lun2p1 /mnt/lun2
# mount
/dev/dasda1 on / type ext4 (rw)
...
/dev/mapper/lun1p1 on /mnt/lun1 type ext4 (rw)
/dev/mapper/lun2p1 on /mnt/lun2 type ext4 (rw)
```

11. Add the following lines to the end of the `/etc/fstab` configuration file to make the new FCP volumes available to the defined mounting points during a server reboot:

```
/dev/mapper/lun1p1       /mnt/lun1               ext4    defaults        0 0
/dev/mapper/lun2p1       /mnt/lun2               ext4    defaults        0 0
```

Congratulations. The FCP disks now are set up and ready for use.

24.5.3 Using FCP with SLES

Adding FCP devices manually is a cumbersome process. With SLES, a YaST module has been created to simplify this process. All of the configurations can also be done with the help of the two scripts **`zfcp_host_configure`** and **`zfcp_disk_configure`**, and in fact, these are also used by YaST. However, reading all the needed WWPNs is somewhat error prone, and thus the description here uses YaST for the configuration.

1. Start an SSN session to the target system.
2. Check that there are two devices available with the **`CP QUERY FCP`** command:

```
# vmcp q v fcp
FCP  0400 ON FCP   BA01 CHPID 7A SUBCHANNEL = 0000
     0400 DEVTYPE FCP         VIRTUAL CHPID 7A FCP REAL CHPID 7A
     0400 QDIO-ELIGIBLE       QIOASSIST-ELIGIBLE
     0400 DATA ROUTER ELIGIBLE
     WWPN C05076E5F9002354
FCP  0500 ON FCP   BB01 CHPID 7B SUBCHANNEL = 0001
     0500 DEVTYPE FCP         VIRTUAL CHPID 7B FCP REAL CHPID 7B
     0500 QDIO-ELIGIBLE       QIOASSIST-ELIGIBLE
     0500 DATA ROUTER ELIGIBLE
     WWPN C05076E5F90024EC
```

3. Run **`yast`** -> `Hardware` -> `zfcp`
4. The following procedure has to be repeated for every possible path to connect each of your FCP host bus adapters (HBAs) to each of the controllers of the storage:
   a. Select Add. A window comes up that preselects one of your devices. Select the first **Channel Number**. In our case, it is 0.0.0400
   b. Select **Get WWPNs** with `[Alt]-G`.
   c. Select the first WWPN by pressing `[Alt]-W` and using the cursor keys and Enter to select.
   d. Select Get LUNs with `[Alt]-E`.
   e. Select the first **FCP-Lun** by pressing [Alt]-F and using the cursor keys and Enter to select. The screen should now look like the following:

```
YaST2 - zfcp @ virtcook6

  Add New ZFCP Device

                Channel Number
                0.0.0400????????????????????????????????????v


                WWPN                             [Get WWPNs]
                0x500507630500c74c???????????????v


                FCP-LUN                           [Get LUNs]
                0x4010401200000000???????????????v
```

f. Click **Next**.

5. Repeat the previous process using all combinations:
   – Use first adapter, first WWPN, second LUN
   – Use first adapter, second WWPN, first LUN
   – Use first adapter, second WWPN, second LUN
   – Use second adapter, first WWPN, first LUN
   – Use second adapter, first WWPN, second LUN
   – Use second adapter, second WWPN, first LUN
   – Use second adapter, second WWPN, second LUN
6. At the end of the procedure, the overview page for zfcp LUNs should look similar to the following:

```
Configured ZFCP Devices
  Minimum Channel                  Maximum Channel
  0.0.0000--------------------------0.f.ffff-------------------------[Filter]
   ----------------------------------------------------------------------

  │Channel Number│              WWPN│          zfcp-LUN                     │
  │      0.0.0400│0x500507630500c74c│0x4010401200000000                     │
  │      0.0.0400│0x500507630500c74c│0x4011401200000000                     │
  │      0.0.0400│0x500507630508c74c│0x4010401200000000                     │
  │      0.0.0400│0x500507630508c74c│0x4011401200000000                     │
  │      0.0.0500│0x500507630500c74c│0x4010401200000000                     │
  │      0.0.0500│0x500507630500c74c│0x4011401200000000                     │
  │      0.0.0500│0x500507630508c74c│0x4010401200000000                     │
  │      0.0.0500│0x500507630508c74c│0x4011401200000000                     │
  │                                                                         │
  │                                                                         │
  │                                                                         │
  │                                                                         │
------------------------------------------------------------------------------
  [Add][Delete]
 [Help]                                            [Cancel]             [Next]
```

7. Click **Next**.
8. Leave YaST with **Quit**.
9. Make sure the `multipath-tools` RPM is installed with the following **zypper** command:

```
# zypper in multipath-tools
```

10. Run the multipath daemon:

```
# chkconfig multipathd on
# rcmultipathd start
```

11. Use YaST to set up the partitioning for the multipath device. In this case, both FCP devices will become a striped XFS for the `/srv/` directory:
    a. Run **yast** -> System -> Partitioner.
    b. Click **Yes** if you are asked if you really want to use this tool.
    c. Select System View -> Hard Disks and press [+].
    d. There are two new devices available that represent the two multipathed FCP disks.
    e. For each FCP device, add a partition that covers the full disk. For both, use **Do not format partition** and **Do not mount partition**.

f. Select System View -> Volume Management.
g. Click Add -> Volume Group.
h. Use srv_vg as Volume Group Name.
i. Add both devices that start with /dev/mapper/360... to the volume group.
j. Click **Finish**.
k. Select System View -> Volume Management.
l. Click Add -> Logical Volume.
m. Set the name of the Logical Volume to srv, click Next.
n. Use **Maximum Size** and change the number of ***Stripes*** to ***2*** and click Next.
o. Select Format partition and use file system XFS.
p. Select **Mount partition** and set the **Mount Point** to /srv.
q. Click finish and Next.
r. The Summary of the changes will look like this:

```
Expert Partitioner: Summary

?????????????????????????????????????????????????????????????????????????????
  ?Changes to partitioning:
?
  ?
?
  ?
?
  ? *  Create multipath partition 36005076305ffc74c0000000000001012_part1
?
  ?    (19.99 GB)
?
  ? *  Create multipath partition 36005076305ffc74c0000000000001112_part1
?
  ?    (19.99 GB)
?
  ? *  Create volume group srv_vg (39.98 GB) from
?
  ?    /dev/mapper/36005076305ffc74c0000000000001012_part1
?
  ?    /dev/mapper/36005076305ffc74c0000000000001112_part1
?
  ? *  Create logical volume /dev/srv_vg/srv (39.98 GB) for /srv with xfs
?
  ?
?
  ?No changes to storage settings.
?
```

s. Click **Finish** and leave YaST with **Quit**.

Check the setup of multipathed devices

FCP is also called *SCSI over Fibre*. Therefore, many of the programs that can be used to debug SCSI can also be used to get information about FCP devices and LUNs:

1. Check if all paths are online:

```
# multipath -ll
36005076305ffc74c0000000000001012 dm-0 IBM     ,2107900
size=20G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='service-time 0' prio=1 status=active
  |- 0:0:0:1074937872 sda 8:0   active ready running
  |- 0:0:1:1074937872 sdc 8:32  active ready running
  |- 1:0:0:1074937872 sdg 8:96  active ready running
  `- 1:0:1:1074937872 sde 8:64  active ready running
36005076305ffc74c0000000000001112 dm-1 IBM     ,2107900
size=20G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='service-time 0' prio=1 status=active
  |- 0:0:0:1074937873 sdb 8:16  active ready running
  |- 0:0:1:1074937873 sdd 8:48  active ready running
  |- 1:0:0:1074937873 sdh 8:112 active ready running
  `- 1:0:1:1074937873 sdf 8:80  active ready running
```

2. Use the **lsscsi** command to list all available SCSI/FCP devices:

```
# lsscsi
[0:0:0:1074937872]disk    IBM      2107900          .120  /dev/sda
[0:0:0:1074937873]disk    IBM      2107900          .120  /dev/sdb
[0:0:1:1074937872]disk    IBM      2107900          .120  /dev/sdc
[0:0:1:1074937873]disk    IBM      2107900          .120  /dev/sdd
[1:0:0:1074937872]disk    IBM      2107900          .120  /dev/sdg
[1:0:0:1074937873]disk    IBM      2107900          .120  /dev/sdh
[1:0:1:1074937872]disk    IBM      2107900          .120  /dev/sde
[1:0:1:1074937873]disk    IBM      2107900          .120  /dev/sdf
```

3. To retrieve vendor-specific information from the storage, use the **sg_inq** command:

```
# sg_inq -p 0x83 /dev/sda
VPD INQUIRY: Device Identification page
  Designation descriptor number 1, descriptor length: 20
    designator_type: NAA,  code_set: Binary
    associated with the addressed logical unit
      NAA 6, IEEE Company_id: 0x5076
      Vendor Specific Identifier: 0x305ffc74c
      Vendor Specific Identifier Extension: 0x1012
      [0x6005076305ffc74c0000000000001012]
  Designation descriptor number 2, descriptor length: 8
    designator_type: Relative target port,  code_set: Binary
    associated with the target port
      Relative target port: 0x3
  Designation descriptor number 3, descriptor length: 8
    designator_type: Target port group,  code_set: Binary
    associated with the target port
      Target port group: 0x0
```

Note, that in this case, the device ID is also available as second line in the **Vendor Specific Identifier Extension**. However, this depends on the storage vendor.

4. To retrieve the information about all eight devices with a single command, use the following:

```
# for dev in /dev/sd{a..h}; do echo $dev ; sg_inq -p 0x83 $dev ; done
```

5. In the s390-tools package, there are several more zfcp-related programs available. Most prominent is probably the program **lszfcp**, which can be used to list all sorts of adapters and connections. To get a full list of zfcp-related programs, use:

```
# rpm -ql s390-tools | grep zfcp
```

Setting up multipath

The multipath daemon has a reasonable default setup. However, it still might be interesting to do some tweaks depending on the local setup. On distributed systems, it might be desirable to blacklist local SCSI devices. This is, however, not expected to happen in a mainframe environment. One major feature that might be interesting is the path grouping and failover behavior, especially if several vendors of FCP storage systems are involved.

For example, if there are two storage types, DS8000 and a NetApp storage system, the following multipath configuration will be appropriate:

```
# vi /etc/multipath.conf
devices {
        device {
                vendor "IBM"
                product ""
                path_grouping_policy group_by_prio
                prio alua
        }
        device {
                vendor "NETAPP"
                product "LUN"
                path_grouping_policy group_by_prio
                prio ontap
        }
}
```

There are many options available to configure different kinds of storages. Many modern FCP storage systems now can be run with Asymmetric Logical Unit Assignment (ALUA). For those that do not support this method, other methods are commonly available. For a complete overview about the configuration options, see the manual page with the **`man 5 multipath.conf`** command.

24.6 HyperPAV for Linux use

HyperPAV in z/VM for Linux guest can be used in several different ways:

- HyperPAV with dedicated DASD

  The Linux system is responsible for managing and serializing I/O requests across the subchannels.

- HyperPAV with minidisks

  The Linux system is not aware of HyperPAV. It will see the minidisk as a regular DASD and it will be able to send only one I/O request at a time to the device. Only when several minidisks are defined on the same real device or when several guests access the same minidisk at the same time, only in those cases HyperPAV would bring any benefit. All those I/O requests will come to z/VM, which will handle them and use HyperPAV aliases as needed.

- HyperPAV minidisks without operating system exploitation

A non-exploiting operating system in a guest is either not configured to use HyperPAV or it cannot use HyperPAV. In such a case, z/VM can utilize HyperPAV on behalf of a non-exploiting guest when several guests share the same full pack minidisk using multiple LINKs.

- HyperPAV minidisks with operating system exploitation

  An exploiting operating system in a guest is able to control HyperPAV features. Such an operating system understands how to control and utilize virtual HyperPAV aliases. Base devices must be defined as full pack minidisks to the guest. Virtual alias devices are then defined using the **DEFINE HYPERPAVALIAS** command.

This section shows how to define HyperPAV to an exploiting Linux operating system.

The following example defines a full pack minidisk at virtual device 102 and six virtual HyperPAV aliases at virtual devices 200-205.

```
USER LINUX3 LNX4VM 512M 1G G
   INCLUDE LNXDFLT
   COMMAND DEFINE HYPERPAVALIAS 200 FOR BASE 102
   COMMAND DEFINE HYPERPAVALIAS 201 FOR BASE 102
   COMMAND DEFINE HYPERPAVALIAS 202 FOR BASE 102
   COMMAND DEFINE HYPERPAVALIAS 203 FOR BASE 102
   COMMAND DEFINE HYPERPAVALIAS 204 FOR BASE 102
   COMMAND DEFINE HYPERPAVALIAS 205 FOR BASE 102
   OPTION APPLMON
   MDISK 0100 3390 0001 5008 JM1268 MR LNX4VM LNX4VM LNX4VM
   MDISK 0101 3390 5009 5008 JM1268 MR LNX4VM LNX4VM LNX4VM
   MDISK 0102 3390 0 END JM1368 MR LNX4VM LNX4VM LNX4VM
```

When the virtual machine is logged on, the disks are defined:

```
...
00: NIC 0600 is created; devices 0600-0602 defined
00: NIC 0600 is connected to VSWITCH SYSTEM VSW1
00: DASD 0200 DEFINED
00: DASD 0201 DEFINED
00: DASD 0202 DEFINED
00: DASD 0203 DEFINED
00: DASD 0204 DEFINED
00: DASD 0205 DEFINED
...
```

When Linux is IPLed, new minidisk and aliases are not available because they were not defined to Linux yet.

```
# lsdasd
Bus-ID     Status      Name      Device  Type  BlkSz  Size      Blocks
==============================================================================
0.0.0100   active      dasda     94:0    ECKD  4096   3521MB    901440
0.0.0301   active      dasdb     94:4    FBA   512    512MB     1048576
0.0.0300   active      dasdc     94:8    FBA   512    256MB     524288
0.0.0101   active      dasdd     94:12   ECKD  4096   3521MB    901440
```

Follow the procedure for adding new disks according to your distribution (edit /etc/dasd.conf for RHEL 6.4 or use **dasd-configure** command in SLES) as described in 24.1, “Add disk space to virtual machines” on page 437.

After devices 102, 200-205 are configured, the output of the **lsdasd** command changes:

```
# lsdasd
Bus-ID     Status      Name      Device  Type  BlkSz  Size      Blocks
==============================================================================
0.0.0200   alias                         ECKD
0.0.0201   alias                         ECKD
0.0.0202   alias                         ECKD
0.0.0203   alias                         ECKD
0.0.0204   alias                         ECKD
0.0.0205   alias                         ECKD
0.0.0100   active      dasda     94:0    ECKD  4096   7042MB    1802880
0.0.0300   active      dasdb     94:4    FBA   512    256MB     524288
0.0.0301   active      dasdc     94:8    FBA   512    512MB     1048576
0.0.0102   active      dasdd     94:12   ECKD  4096   7043MB    1803060
```

There is no other configuration needed. From now on, whenever /dev/dasdd is used, Linux will use the base device as well as alias devices to distribute the workload. There is no multipathing needed for HyperPAV to work in an exploiting Linux.

If another device is added which comes from the same LSS as device 102 above, it will also use the same virtual HyperPAV devices. Only if a real device from different LSS is added, it will need a new set of virtual HyperPAV aliases added. Keep in mind the following when defining virtual HyperPAV aliases:

- Base device has to be defined as a full pack minidisk.
- Because base device is defined as a full pack minidisk, Linux has control of cylinder 0 as well. It means **dasdfmt** will overwrite the volume serial with 0xyyyy where yyyy is the virtual device number. To solve this issue, use **-l** parameter with **dasdfmt** to set the correct volume serial and **-k** parameter with **fdasd** to preserve a volume serial.
- The sum of virtual HyperPAV aliases defined in one guest for one LSS cannot be higher that the number of real aliases in that LSS's pool of aliases. It does not matter if all virtual HyperPAV aliases in the guest are defined to one base device or they are spread among several devices, they all act as one pool of aliases for a given LSS. For example, assume that there is an LSS with 20 real devices and 20 aliases in a HyperPAV pool. We want to define four Linux images, each with five real devices. To configure each Linux for the maximum possible throughput, we should define each Linux with 20 virtual HyperPAV aliases. They can be defined to one real device or spread among all real devices, result would be the same, there will be 20 aliases for five devices in a guest:

```
USER LINUX1...
...
COMMAND DEFINE HYPERPAVALIAS 200 FOR BASE 100
...
COMMAND DEFINE HYPERPAVALIAS 213 FOR BASE 100
MDISK 100 3390 0 END VOL001 MR
...
MDISK 104 3390 0 END VOL005 MR
```

would achieve the same 20 aliases for five devices as the following definition. The former is obviously easier to read.

```
USER LINUX2...
...
COMMAND DEFINE HYPERPAVALIAS 200 FOR BASE 100
COMMAND DEFINE HYPERPAVALIAS 201 FOR BASE 100
...
COMMAND DEFINE HYPERPAVALIAS 204 FOR BASE 101
```

```
COMMAND DEFINE HYPERPAVALIAS 205 FOR BASE 101
...
COMMAND DEFINE HYPERPAVALIAS 208 FOR BASE 102
COMMAND DEFINE HYPERPAVALIAS 209 FOR BASE 102
...
COMMAND DEFINE HYPERPAVALIAS 213 FOR BASE 104
MDISK 100 3390 0 END VOL001 MR
...
MDISK 104 3390 0 END VOL005 MR
```

More about HyperPAV in z/VM is described in detail in *z/VM CP Planning and Administration, SC24-6178*.

25

Working with networks

"*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:

25.1 Attach the z/VM TCP/IP stack to the VSWITCH

Section 5.4.1, "Use the IPWIZARD tool" on page 72 describes how to set up TCP/IP on z/VM and attach real OSA triplets using the **IPWIZARD** tool. This quickly gets new z/VM systems onto the network. Then, a VSWITCH was created with failover OSA addressees. It is recommended that the z/VM stack now be attached to the VSWITCH so that it does not have OSA devices as a single point of failure. To do this, perform the following steps:

1. On single system image (SSI) member 1, log on to TCPMAINT.
2. Edit the SYSTEM DTCPARMS file on the TCPMAINT 198 (D) disk.
3. Comment out the last line with a **.*** in the first two columns so the OSA triplet will *not* be be attached to the TCP/IP virtual machine:

```
==> x system dtcparms d
.**********************************************************************
.* SYSTEM DTCPARMS created by DTCIPWIZ EXEC on 19 Apr 2012
.* Configuration program run by MAINT at 09:56:59
.**********************************************************************
:nick.TCPIP    :type.server
               :class.stack
.*             :attach.2100-2102
```

   This change will prevent three OSA devices from being attached to TCPIP as it is started.

4. Make a backup copy of the working PROFILE TCPIP file created by the **IPWIZARD**:

```
==> copy profile tcpip d = tcpiwrks =
```

5. Edit the PROFILE TCPIP file on the TCPMAINT 198 (D) disk. Change the real OSA starting address (***2100*** in this example) to the virtual starting address (0600) everywhere in the file:

```
==> x profile tcpip d
====> c/2100/0600/* *
DMSXCG517I 4 occurrence(s) changed on 3 line(s)
```

   This will instruct TCPIP to use the virtual NIC starting at virtual device address 600.

6. Log off from TCPMAINT.
7. Log on to MAINT.
8. If you are not using a directory maintenance product such as DirMaint, make a copy of the original USER DIRECT file on the MAINT 2CC (C) disk:

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

   If you are using DirMaint, you would edit the IDENTITY TCPIP with the command DIRM FOR TCPIP GET.

9. Edit the USER DIRECT file and locate the TCPIP definition. Add the following three lines that will connect the virtual machine to the VSWITCH VSW1 at logon time:

```
==> x user direct c
====> /identity tcpip
...
IDENTITY TCPIP    TCPIP     128M  256M ABG
 INCLUDE TCPCMSU
 BUILD ON ZVM63A USING SUBCONFIG TCPIP-1
 BUILD ON ZVM63B USING SUBCONFIG TCPIP-2
* BUILD ON @@member3name USING SUBCONFIG TCPIP-3
* BUILD ON @@member4name USING SUBCONFIG TCPIP-4
 COMMAND SET VSWITCH VSW1 GRANT &USERID
```

```
 COMMAND DEFINE NIC 600 TYPE QDIO
 COMMAND COUPLE 600 TO SYSTEM VSW1
...
====> file
```

10. These statements will grant TCP/IP access to VSWITCH `VSW1`, define a virtual NIC starting at virtual device address `600`, and couple it to the VSWITCH.
11. Run the **`DIRECTXA`** command to put the changes online:

```
==> directxa user
z/VM USER DIRECTORY CREATION PROGRAM - VERSION 6 RELEASE 3.0
EOJ DIRECTORY UPDATED AND ON LINE
HCPDIR494I User directory occupies 107 disk pages
```

12. **Repeat the previous steps in this section** on all other members in the SSI, however, the three **`COMMAND`** lines added to the `IDENTITY TCPIP` should already be present, so the XEDIT step is unnecessary this time.

> **Note:** If RACF is enabled on your system, invoke the following two commands:
>
> ```
> RAC PERMIT SYSTEM.vsw1 CLASS(VMLAN) ID(tcpip)ACCESS(UPDATE)
> RAC SETROPTS CLASSACT(VMLAN)
> ```

The z/VM TCP/IP stack should come up on the highly available VSWITCH the next time z/VM is IPLed.

25.2 Adding CTCs to an SSI cluster

The SSI CTC install panel allows two CTC connections to be installed per SSI member. You should add CTCs for performance and redundancy. It is recommended that four out of the eight CTC devices be used to connect SSI members via each channel path. Generally, eight devices will be available in a FICON CTC control unit. It is recommended for performance reasons that only four out of the eight devices are used.

The following example will add three CTCs per member to each path that was activated during the installation.

- Display the installed CTCs on the first member (`ZVM63A`:

```
==> q ctc active
CTCA 47E0 ATTACHED TO SYSTEM  -ISFC
CTCA 57E0 ATTACHED TO SYSTEM  -ISFC
```

- Display the installed CTCs on `ZVM63B`:

```
==> q ctc active
CTCA 4120 ATTACHED TO SYSTEM  -ISFC
CTCA 5120 ATTACHED TO SYSTEM  -ISFC
```

The previous two commands show the four CTCs that were set up during z/VM installation. From these real device addresses, determine the channel paths that they are on with the following commands:

- Display the channel paths used by the CTCs on `zVM63A`:

```
==> q path to 47e0
Device 47E0, Status ONLINE
CHPIDs to Device 4120 (PIM)  : 4C
```

```
==> q path to 57e0
Device 57E0, Status ONLINE
CHPIDs to Device 57E0 (PIM)  : 4D
```

- Display the channel paths used by the CTCs on ZVM63B:

```
==> q path to 4120
Device 4120, Status ONLINE
CHPIDs to Device 4120 (PIM)  : 4C
==> q path to 5120
Device 5120, Status ONLINE
CHPIDs to Device 5120 (PIM)  : 4D
```

The previous two commands show the CHPIDs that the CTCs are on. In this example, they are 4C and 4D. From these CHPIDs, determine the other CTC devices available by doing the following:

- Display the devices used by the channel paths on ZVM63A:

```
==> q chpid 4c
Path 4C online to devices 47E0 47E1 47E2 47E3 4A90 4A91 4A92 4A93
==> q chpid 4d
Path 4D online to devices 57E0 57E1 57E2 57E3 5A90 5A91 5A92 5A93
```

- Display the devices used by the channel paths on zVM63B:

```
==> q chpid 4c
Path 4C online to devices 4120 4121 4122 4123 4A90 4A91 4A92 4A93
==> q chpid 4d
Path 4D online to devices 5120 5121 5122 5123 5A90 5A91 5A92 5A93
```

It is recommended to confirm with your hardware configuration engineer that you can add three CTCs to each channel path on each z/VM member. They should be added both dynamically and permanently. Next, you will run the following commands:

- Verify that the next three CTCs are available on zVM63A:

```
==> q 47e1 47e2 47e3
CTCA 47E1 FREE    , CTCA 47E2 FREE    , CTCA 47E3 FREE
==> q 57e1 57e2 57e3
CTCA 57E1 FREE    , CTCA 57E2 FREE    , CTCA 57E3 FREE
```

- Verify that the next three CTCs are available on zVM63B:

```
==> q 4121 4122 4123
CTCA 47E1 FREE    , CTCA 47E2 FREE    , CTCA 47E3 FREE
==> q 5121 5122 5123
CTCA 5121 FREE    , CTCA 5122 FREE    , CTCA 5123 FREE
```

You should now have the real device addresses of the CTCs to be added to each SSI member.

25.2.1 Add the CTC devices dynamically

To add the CTC devices dynamically, perform the following steps:

1. Log on to MAINT on the first member:
2. Activate six CTCs on the first member zVM63A:

```
==> activate islink 47e1 47e2 47e3 57e1 57e2 57e3
Link device 47E1 activated.
Link device 47E2 activated.
```

```
Link device 47E3 activated.
Link device 57E1 activated.
Link device 57E2 activated.
Link device 57E3 activated.
```

3. Activate six CTCs on zVM63B:

```
==> activate islink 4121 4122 4123 5121 5122 5123
Link device 4121 activated.
Link device 4122 activated.
Link device 4123 activated.
Link device 5121 activated.
Link device 5122 activated.
Link device 5123 activated.
```

When the device is active on both systems, you will see a **HCPKCL2714I** message. You will see the additional CTCs if you reissue the **QUERY CTC** command.

4. Issue the **QUERY CTC** command from ZVM63A:

```
==> q ctc
CTCA 47E0 ATTACHED TO SYSTEM  -ISFC
CTCA 47E1 ATTACHED TO SYSTEM  -ISFC
CTCA 47E2 ATTACHED TO SYSTEM  -ISFC
CTCA 47E3 ATTACHED TO SYSTEM  -ISFC
CTCA 57E0 ATTACHED TO SYSTEM  -ISFC
CTCA 57E1 ATTACHED TO SYSTEM  -ISFC
CTCA 57E2 ATTACHED TO SYSTEM  -ISFC
CTCA 57E3 ATTACHED TO SYSTEM  -ISFC
```

5. Issue the **QUERY CTC** command from ZVM63B:

```
==> q ctc
CTCA 4120 ATTACHED TO SYSTEM  -ISFC
CTCA 4121 ATTACHED TO SYSTEM  -ISFC
CTCA 4122 ATTACHED TO SYSTEM  -ISFC
CTCA 4123 ATTACHED TO SYSTEM  -ISFC
CTCA 5120 ATTACHED TO SYSTEM  -ISFC
CTCA 5121 ATTACHED TO SYSTEM  -ISFC
CTCA 5122 ATTACHED TO SYSTEM  -ISFC
CTCA 5123 ATTACHED TO SYSTEM  -ISFC
```

This shows that the CTC devices have been added dynamically.

25.2.2 Add the CTC devices permanently

To add the CTC devices to the SSI permanently, perform the following steps:

1. Log on to MAINT on the first SSI member.
2. Access the PMAINT CF0 disk read/write and link as file mode F:

```
==> link pmaint cf0 cf0 mr
==> acc cf0 f
```

3. Make a backup copy of the SYSTEM CONFIG file:

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

4. Edit the SYSTEM CONFIG file and find the ISLINK statements with **/Activate ISLINK** subcommand. Change ISLINK statements to include the new CTCs. BEGIN and END statements are added as the new values require two lines each:

```
==> x system config f
====> /activate islink
```

The following shows the SYSTEM CONFIG file before and after the changes are made.

Before:

```
/**********************************************************************/
/*         Activate ISLINK statements                                 */
/**********************************************************************/

ZVM63A:      ACTIVATE ISLINK 47E0 57E0  NODE ZVM63B
ZVM63B:      ACTIVATE ISLINK 4120 5120  NODE ZVM63A
```

After:

```
/**********************************************************************/
/*         Activate ISLINK statements                                 */
/**********************************************************************/

ZVM63A:      BEGIN
   ACTIVATE ISLINK 47E0 47E1 47E2 47E3  NODE ZVM63B
   ACTIVATE ISLINK 57E0 57E1 57E2 57E3  NODE ZVM63B
ZVM63A:      END
ZVM63B:      BEGIN
   ACTIVATE ISLINK 4120 4121 4122 4123  NODE ZVM63A
   ACTIVATE ISLINK 5120 5121 5122 5123  NODE ZVM63A
ZVM63B:      END
```

When the system is restarted, the ISLINKs will be active between members.

25.3 Setting up a private interconnect

It can be beneficial to have a means of networked communications between different hosts that belong to a certain group. For example, some legal database will want to *talk* to machines that scan documents for legal issues. Or a web server and some backend machine want to talk to each other without other machines interfering. In times before live relocation, it was sufficient to just set up a VSWITCH without an external interface to accomplish these tasks.

However, when trying to run this interconnect between hosts that run on a cross-CEC SSI cluster, the private interconnect must have some means to connect the network on the respective guests. An easy way to accomplish this requirement is to set up a VLAN for each of the required private interconnects on the external network. For each of these VLANs, then create a VLAN-aware VSWITCH with PORTTYPE access. To accomplish this, proceed as follows:

1. Set up a network switch that connects to the mainframe and configure all needed VLANs as tagged VLANs to the attached port.
2. Find a free port triplet on the OSA device, for example, for the devices ***903-905***.

3. Edit the system configuration and add the following statement to the end of the file:

```
DEFINE VSWITCH PRV01 RDEV 0903 ETH VLAN 75 PORTT ACCESS
```

4. Grant only the group of virtual machines access that should be on that network:

```
MODIFY VSWITCH PRV01 GRANT LINUXADM
MODIFY VSWITCH PRV01 GRANT LINUX5
```

5. Do the same on all other members of the SSI.
6. Define a private IP range for the group of hosts. It is good practice to track the IP ranges and not overlap them, even if the respective hosts do not have a network connection between them.

More specific details and a working implementation were wanted, however, the time to write this section was not sufficient.

25.4 Creating a HiperSockets device between Linux and z/OS

IBM HiperSockets™ devices can be used within a CEC to enable fast and secure connectivity between a Linux server and z/OS. This section describes how to do the following actions:

- Verify HiperSockets hardware definitions
- Create a TCP/IP stack on z/OS
- Verify HiperSockets hardware definitions
- Verify connectivity

25.4.1 Verify HiperSockets hardware definitions

Connectivity requires a HiperSockets IQD CHPID and devices that can be accessed by both the z/OS LPAR and the Linux z/VM LPAR. You will see in Figure 2-2 on page 20 that we defined a HiperSockets connection CHPID `F0` between z/OS LPAR A12 and z/VM LPAR A02 using devices `7000`. This diagram is defined in the IOCP statements that follow:

```
CHPID PATH=(CSS(0,1,2,3),F0),SHARED,                         *
      NOTPART=((CSS(0),(A04,A0C,A0D,A0E,A0F),(=)),(CSS(1),(A1B*
      ,A1D,A1E,A1F),(=)),(CSS(2),(A2E,A2F),(=)),(CSS(3),(A32,A*
      3D,A3E,A3F),(=))),TYPE=IQD
CNTLUNIT CUNUMBR=7000,                                       *
      PATH=((CSS(0),F0),(CSS(1),F0),(CSS(2),F0),(CSS(3),F0)), *
      UNIT=IQD
IODEVICE ADDRESS=(7000,16),UNITADD=00,CUNUMBR=(7000),UNIT=IQD
```

You will notice that VM LPAR A02 and z/OS LPAR A12 have access to the hipersocket CHPID F0 and it is an IQD type.

25.4.2 Create a TCP/IP stack on z/OS

To create a TCP/IP stack within z/OS to use the HiperSockets device, it is recommended to get assistance from your network team. More information at HiperSockets connectivity is available within the IBM Redbooks publication HiperSockets implementation guide, which can be found at the following website:

http://www.redbooks.ibm.com/redbooks/pdfs/sg246816.pdf

- Create a TCP/IP stack (called `TCPIPF` in this example) with a TCP/IP profile that uses the ***F0*** CHPID:

```
VIEW       TCPIPF.SC42.TCPPARMS(TCPPROF) - 01.05
Command ===>
000085
000086 DEVICE IUTIQDFO MPCIPA
000087 LINK   HIPERLFO    IPAQIDIO      IUTIQDFO
000088
...
000090 HOME
000093    10.1.1.42     HIPERLFO
..
000097 BEGINROUTES
..
000102 ROUTE 10.1.1.0 255.255.255.0 = HIPERLFO MTU 8192
000103 ENDROUTES
000104
000107 START IUTIQDFO
```

Put the CHPID identifier within the IUTIQDxx device statement. If it meets your sites' requirements, place the CHPID identifier in the LINK statements. Give the link a `HOME` address and `ROUTE` address according to your site networking requirements. Start your **`TCPIPF`** address space that uses this profile

Issue the command **`D TCPIP,TCPIPF,NETSTAT,DEVL`** to verify link information.

25.4.3 Creating a TCP/IP stack on Linux

In this section, we describe how to create a TCP/IP stack on Linux:

1. From the Linux virtual machine, verify that the real devices are available:

```
==> q chpid f0
Path F0 online to devices 7000 7001 7002 7003 7004 7005 7006 7007
Path F0 online to devices 7008 7009 700A 700B 700C 700D 700E 700F
==> q 7000-7002
OSA  7000 FREE    , OSA  7001 FREE    , OSA  7002 FREE
```

2. Attach the hipersocket devices to the Linux image using virtual device numbers. The command is issued from Linux1 in this example:

```
==> attach 7000 e000
OSA 7000 ATTACHED TO LINUX1 E000
==> attach 7001 e001
OSA 7001 ATTACHED TO LINUX1 E001
==> attach 7002 e002
OSA 7002 ATTACHED TO LINUX1 E002
```

3. Verify the HiperSockets device type:

```
==> q 7000-7002
OSA  7000 ATTACHED TO LINUX1 E000 DEVTYPE HIPER       CHPID F0 IQD
OSA  7001 ATTACHED TO LINUX1 E001 DEVTYPE HIPER       CHPID F0 IQD
OSA  7002 ATTACHED TO LINUX1 E002 DEVTYPE HIPER       CHPID F0 IQD
```

4. Make the changes permanent with the following DIRM commands:

```
==> DIRM FOR LINUX1 DEDICATE E800 7000
==> DIRM FOR LINUX1 DEDICATE E801 7001
==> DIRM FOR LINUX1 DEDICATE E802 7002
```

Red Hat Enterprise Linux

If you are using Red Hat Enterprise Linux, perform the following steps to create the device:

1. From the Linux image, create a device group for the E800 devices:

```
# echo 0.0.E800,0.0.E801,0.0.E802 > /sys/bus/ccwgroup/dirvers/qeth/group
```

2. Bring the device online:

```
# echo 1 > /sys/devices/qeth/0.0.e800/online
```

3. Get the name of the devices from command:

```
# cat /sys/devices/qeth/0.0.e800/if_name
   hsi0
```

4. Create a network configuration file using the name found in `if_name` in the location `/etc/sysconfig/network-scripts/`:

```
# vi /etc/sysconfig/network-scripts/ifcfg-hsi0
#IBM QETH
DEVICE=hsi0
BOOTPROTO=static
IPADDR=10.1.1.43
NETMASK=255.255.255.0
NETTYPE=qeth
ONBOOT=yes
SUBCHANNELS=0.0.E800,0.0.E801,0.0.E802
TYPE=ethernet
ARP=no
```

5. Start the `hsi0` network device with the `ifup` command:

```
==> ifup hsi0
```

6. Verify the `hsi0` status with the **ifconfig** command:

```
==> ifconfig hsi0
hsi0      Link encap:Ethernet  HWaddr 06:00:F0:02:00:02
          inet addr:10.1.1.43  Bcast:10.1.1.255  Mask:255.255.255.0
          inet6 addr: fe80::400:f0ff:fe02:2/64 Scope:Link
          UP BROADCAST RUNNING NOARP MULTICAST  MTU:8192  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:26 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:2296 (2.2 KiB)
```

SUSE Linux Enterprise Server

If you are using the SUSE Linux Enterprise Server distribution of Linux, perform the following steps:

1. Configure the second NIC with the **qeth_configure** command:

```
# qeth_configure -t hsi 0.0.7000 0.0.7001 0.0.7002 1
```

2. As `hsi0` was created `/etc/sysconfig/network/ifcfg-hsi0`

```
# vi /etc/sysconfig/network/ifcfg-hsi0
BOOTPROTO='static'
IPADDR='10.1.1.46/24'
STARTMODE='onboot'
NAME='HIPERSOCKETS (0.0.7400)'
```

3. Bring the `hsi0` device up with the **ifup** command:

```
# ifup hsi0
```

The HiperSockets device should now be up.

25.4.4 Verify connectivity

To verify that the HiperSockets device is functioning, perform the following steps:

1. Ping from z/OS UNIX Systems Services:

   ```
   USER1 @ SC42:/u/user1>ping 10.1.1.43
   CS V1R13: Pinging host 10.1.1.43
   Ping #1 response took 0.000 seconds.
   ```

2. Ping from the Red Hat Enterprise Linux running on ZVM63A:

   ```
   [root@virtcook1 etc]# ping 10.1.1.42
   PING 10.1.1.42 (10.1.1.42) 56(84) bytes of data.
   64 bytes from 10.1.1.42: icmp_seq=1 ttl=64 time=0.025 ms
   ```

3. Ping from SUSE Linux Enterprise Server running on ZVM63B:

   ```
   linuxadmin:/etc/sysconfig/network # ping 10.1.1.46
   PING 10.1.1.46 (10.1.1.46) 56(84) bytes of data.
   ```

This shows that the hipersocket is working.

25.5 Configuring a port group with LACP

To aggregate multiple OSA-Express ports, *port groups* can be defined on z/VM and attached to a virtual switch. Connectivity using a port group requires OSA devices that are only to be used by one z/VM LPAR. This example uses four-port OSA express cards, which use two ports per CHPID. See Figure 25-1.

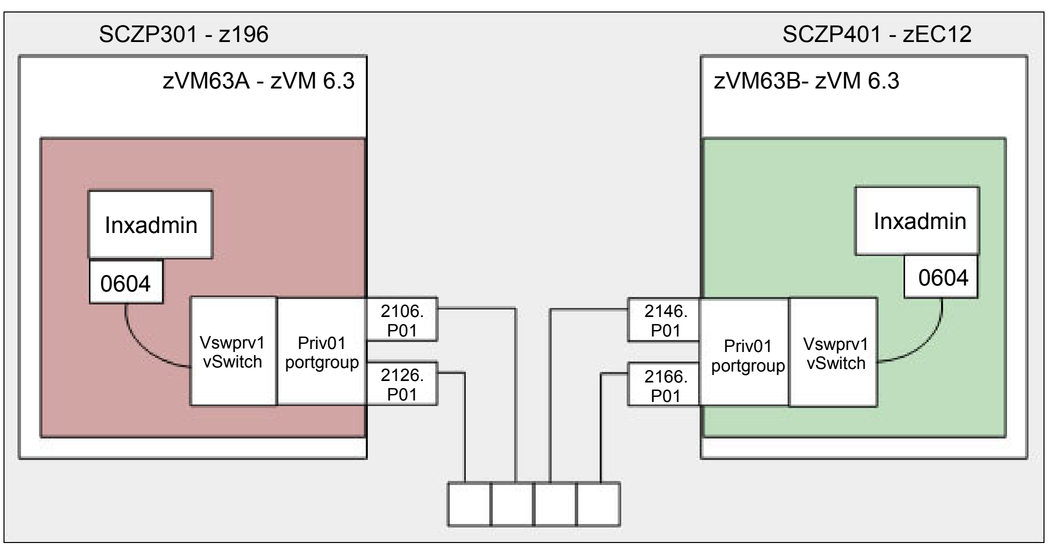


Figure 25-1 Port group Priv01 connectivity

Note: Port number 1 not port number 0 was used for this connection.

zVM63A port group priv1 details:

```
CHPID 00 portnumber 1 OSA device 2106
CHPID 01 portnumber 1 OSA device 2126
```

zVM63B port group priv1 details:

```
CHPID 00 portnumber 1 OSA device 2046
CHPID 01 portnumber 1 OSA device 2066
```

You will use the following steps to accomplish this task:

1. Create the port group on the first SSI member (`ZVM63A` in this example):

   `==>` **`set port group priv01 join 2106.p01 2126.p01`**

2. Create the port group on the second SSI member (`ZVM63B` in this example):

   `==>` **`set port group priv01 join 2046.p01 2066.p01`**

   > **Note:** LCAP is set active by default. To use LACP, the network switch will need LACP active on the ports that the CHPIDs connect to.

   Important: If you receive message `HCPSWU2832E`, the LPAR does not have exclusive use of the device. Another LPAR has the device online. However, exclusive use does not require that only one LPAR in the IOCDS has the CHPID defined and the devices dedicated.

3. Define the virtual switch using the `priv01` portgroup on all members of the SSI (`ZVM63A` and `ZVM63B` in this example):

   `==>` **`define vswitch vswprv1 rdev none ethernet vlan aware group priv01 gvrp`**

4. Create a virtual NIC on the Linux image to use the switch. If you are running DirMaint, the following command will create a network device and add it to `LNXADMIN`:

   `==>` **`dirm for lnxadmin nicdef 604 type qdio lan system vswprv1`**

5. If you are not running a directory maintenance product, you will need to update the user directory entry for `LNXADMIN` with the following **`DEFINE NIC`** and **`COUPLE`** commands:

   ```
   COMMAND SET VSWITCH VSWPRV1 GRANT &USERID
   COMMAND DEFINE NIC 604 TYPE QDIO
   COMMAND COUPLE 604 TO SYSTEM VSWPRV1
   ```

6. If you have RACF running without DIRM, you will need to permit the virtual switch `VSWPRV1` access to `LNXADMIN` with the following commands:

   a. Define the system virtual switch named `VSWPRV1` to the `VMLAN` class:

      `==>` **`rac rdefine vmlan system.vswprv1`**

   b. Permit TCPIP to the virtual switch `VSW1`:

      `==>` **`rac permit system.vswprv1 class(vmlan) id(lnxadmin) access(update)`**

Red Hat Enterprise Linux

If you are on a Red Hat Enterprise Linux system, perform the following steps to create the network device `ETH1`:

1. From the Linux image, create a device group for the `0604` devices:

   `#` **`echo 0.0.0604,0.0.0605,0.0.0606 > /sys/bus/ccwgroup/drivers/qeth/group`**

2. Bring the device online with the following command:

   `#` **`echo 1 > /sys/devices/qeth/0.0.0604/online`**

3. Get the name of the device:

```
# cat /sys/devices/qeth/0.0.0604/if_name
eth1
```

4. Create a network configuration file using the name `eth1` in the file: `/etc/sysconfig/network-scripts/ifcfg-eth1`:

```
==> vi /etc/sysconfig/network-scripts/ifcfg-eth1
#IBM QETH
DEVICE=eth1
BOOTPROTO=static
IPADDR=10.1.1.47
NETMASK=255.255.255.0
NETTYPE=qeth
ONBOOT=yes
SUBCHANNELS=0.0.0604,0.0.0605,0.0.0606
TYPE=ethernet
ARP=no
```

5. Start the eth1 network device with the **ifup** command:

```
==> ifup eth1
```

6. Verify the status of `eth1` with the **ifconfig** command:

```
==> ifconfig eth1
eth1      Link encap:Ethernet  HWaddr 02:00:0B:00:00:0B
          inet addr:10.1.1.47  Bcast:10.1.1.255  Mask:255.255.255.0
          inet6 addr: fe80::bff:fe00:b/64 Scope:Link
          UP BROADCAST RUNNING NOARP MULTICAST  MTU:1492  Metric:1
          RX packets:8 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2464 (2.4 KiB)  TX bytes:350 (350.0 b)
```

SUSE Linux Enterprise Server

If you are on a SUSE Linux Enterprise Server system, perform the following steps to create the network device `ETH1`:

1. Run the following command to create the device on `LNXADMIN`:

```
# qeth_configure -l -t qeth 0.0.0604 0.0.0605 0.0.0606 1
```

2. Create the interface `eth1` using the file `/etc/sysconfig/network/ifcfg-eth1`:

```
# vi /etc/sysconfig/network/ifcfg-eth1
BOOTPROTO='static'
IPADDR='10.1.1.48/24'
STARTMODE='onboot'
NAME='OSA Express(0.0.0604)'
```

3. Open the `eth1` device with the `ifup` command:

```
# ifup eth1
```

4. Test connectivity between each Linux image.

You should now have a functioning network interface utilizing port groups.

26

Miscellaneous recipes

"*Try not to become a man of success, but rather try to become a man of value.*"

— Albert Einstein

This chapter has the following sections of miscellaneous tasks that you might want to perform:

26.1 Rescue 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.

26.1.1 Enter 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 what is shown here:

```
zIPL v1.8.2-28.el6 interactive boot menu

 0. default (linux)
```



```
 1. linux

Note: VM users please use '#cp vi vmsg <input>'

Please choose (default will boot in 5 seconds):
```

You can use the **#cp vi vmsg** command to boot the wanted menu option (zero in this example), followed by the number one for single user mode. After you log in to Linux from the 3270 emulator, you can use the **runlevel** command to verify the run level that you are in:

```
==> #cp vi vmsg 0 1
...
# runlevel
1 S
```

In single user mode, you are logged in as the root user. You can use the **passwd** command to set the `root` password. All of the file systems in `/etc/fstab` are mounted, but networking has not been started. To exit single user mode, you can type **reboot**, or enter **telinit 3** to continue booting normally.

26.1.2 Enter 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.

Using rescue mode with RHEL

To enter a rescue environment, initiate an interactive Linux installation. Perform the following steps to enter a rescue environment on the `RH64GOLD` virtual machine:

1. Log on to `LNXMAINT`. Copy the **RHEL64 EXEC** file to a new file named **RESCUE EXEC**, and copy the user's `PARM-RH6` file to a new file (`RH64GOLD RESCUE` in this example):

   ```
   ==> copy rhel64 exec d rescue = =
   ==> copy rh64gold parm-rh6 d = rescue =
   ```

2. Edit `RESCUE EXEC` to point to the new `RESCUE` file:

   ```
   ==> x rescue exec d
   ...
   Address 'COMMAND'
     'CP SPOOL PUN *'
     'CP CLOSE RDR'
     'CP PURGE RDR ALL'
     'PUNCH RHEL64 KERNEL * (NOHEADER'
     'PUNCH' 'USERID'() 'RESCUE   * (NOHEADER'
     'PUNCH RHEL64 INITRD * (NOHEADER'
     'CP CHANGE RDR ALL KEEP'
     'CP IPL 00C CLEAR'
     Exit
   ```

3. Edit the `RH64GOLD RESCUE` file, replacing any kickstart or VNC lines with the **rescue** command-line option:

   ```
   ==> x rh64gold rescue d
   root=/dev/ram0 ro ip=off ramdisk_size=40000
   CMSDASD=191 CMSCONFFILE=RH64GOLD.CONF-RH6
   rescue
   ```

4. Log off from `LNXMAINT`.
5. Log on to `RH64GOLD` and answer **no** to IPL from `100` question.

6. Define storage to 1 GB.

```
==> def stor 1g
00: STORAGE = 1G
00: Storage cleared - system reset.
Enter
```

7. IPL CMS:

```
==> ipl cms
```

8. Run the **RESCUE EXEC**:

```
==> rescue
00:      NO FILES PURGED
00: RDR FILE 0001 SENT FROM RH62GOLD PUN WAS 0001 RECS 113K CPY  001 A NOHOLD NO
KEEP
00: RDR FILE 0005 SENT FROM RH62GOLD PUN WAS 0005 RECS 0003 CPY  001 A NOHOLD NO
KEEP
00: RDR FILE 0009 SENT FROM RH62GOLD PUN WAS 0009 RECS 224K CPY  001 A NOHOLD NO
KEEP
00: 0000003 FILES CHANGED
00: 0000003 FILES CHANGED
...
Kernel command line: root=/dev/ram0 ro ip=off ramdisk_size=40000
                     CMSDASD=191 CMSCONFFILE=RH64GOLD.CONF-RH6
                     rescue
...
Starting sshd to allow login over the network.

Connect now to 9.60.18.144 and log in as user 'install' to start the installatio
n.
E.g. using: ssh -x install@9.60.18.144
For VNC or text mode, disable X11 forwarding (recommended) with 'ssh -x'.
For X11, enable X11 forwarding with 'ssh -X'.

You may log in as the root user to start an interactive shell.
```

The installation process directs you to Telnet or Secure Shell (SSH) to the IP address of your Linux server to begin the first stage of the installation.

9. Use SSH to connect to the IP address and log in as **install**.

10. You should get an SSH session and see a *Choose a Language* panel. Choose your language.

11. The rescue environment will prompt you for the location of the rescue image, which is located in the installation tree on the Linux administration system. Choose **NFS directory**, then enter the IP address of the Linux administration system and the path **/var/nfs/rhel64**.

```
+------------------------------┤ NFS Setup ├------------------------------+
¦                                                                         ¦
¦ Please enter the server and NFSv3 path to your Red Hat Enterprise Linux ¦
¦ installation image and optionally additional NFS mount options.         ¦
¦                                                                         ¦
¦     NFS server name:                   9.12.7.7_____________            ¦
¦     Red Hat Enterprise Linux directory: /var/nfs/rhel64_________        ¦
```

12. The *Rescue* window appears. Choose **Continue**. The rescue image will search for your Linux installation.

13. Hopefully, it will prompt you to mount the partitions that it finds:

```
+--------------¦ Rescue +---------------+
¦                                       ¦
¦ Your system has been mounted under    ¦
¦ /mnt/sysimage.                        ¦
¦                                       ¦
¦ Press <return> to get a shell. If you ¦
¦ would like to make your system the    ¦
¦ root environment, run the command:    ¦
¦                                       ¦
¦         chroot /mnt/sysimage          ¦
¦                                       ¦
¦ The system will reboot automatically  ¦
¦ when you exit from the shell.         ¦
```

Note: If the rescue image cannot find your partition, you can try to mount it yourself with 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
```

Note: Type **exit** to leave the shell and exit rescue mode.

Using rescue mode with SUSE Linux Enterprise Server

The procedure to enter rescue mode in SLES is very similar to a manual installation. To enter rescue mode for LINUX6 in this example, proceed as follows:

1. Log on to LNXMAINT.
2. Copy the **SLES11S3 EXEC** file to a new file named **RESCUE EXEC**, and copy the parameter file to a new file (LINUX6 RESCUE in this example):

```
==> copy sles11s3 exec d rescue = =
==> copy linux6 parm-s11 d = rescue =
```

3. Edit RESCUE EXEC to point to the new RESCUE file:

```
==> x rescue exec d
...
Address 'COMMAND'
  'CP SPOOL PUN *'
  'CP CLOSE RDR'
  'CP PURGE RDR ALL'
  'PUNCH SLES11S3 KERNEL * (NOHEADER'
  'PUNCH' 'USERID'() 'RESCUE   * (NOHEADER'
  'PUNCH SLES11S3 INITRD * (NOHEADER'
  'CP CHANGE RDR ALL KEEP'
  'CP IPL 00C CLEAR'
  Exit
```

4. Edit the LINUX6 RESCUE file, deleting the autoyast line and set manual=1:

```
==> x linux6 rescue d
LINUX6   RESCUE   D1  F 80  Trunc=80 Size=10 Line=0 Col=1 Alt=1

ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc
```

```
TERM=dump HostIP=9.12.7.6 Hostname=virtcook6.itso.ibm.com
Gateway=9.12.4.1 Netmask=255.255.240.0 Layer2=1
ReadChannel=0.0.0600 WriteChannel=0.0.0601
DataChannel=0.0.0602 OSAHWAddr=
Nameserver=9.12.6.7 Portname= Portno=0
Install=ftp://9.12.7.8/SLES-11-SP3
UseVNC=1 VNCPassword=12345678
UseSSH=1 SSHPassword=12345678
InstNetDev=osa OsaInterface=qdio OsaMedium=eth Manual=1
```

5. Log off from `LNXMAINT`.
6. Log on to `LINUX6`.
7. Answer **no** to IPL from `100` question.
8. Define storage to 1 GB.

```
==> def stor 1g
00: STORAGE = 1G
00: Storage cleared - system reset.
Enter
```

9. IPL CMS:

```
==> ipl cms
```

10. Run the **RESCUE EXEC**:

```
==> rescue
00: 0000001 FILE  PURGED
00: RDR FILE 0002 SENT FROM LINUX6   PUN WAS 0002 RECS 113K CPY  001 A NOHOLD NO
KEEP
00: RDR FILE 0006 SENT FROM LINUX6   PUN WAS 0006 RECS 0011 CPY  001 A NOHOLD NO
KEEP
00: RDR FILE 0010 SENT FROM LINUX6   PUN WAS 0010 RECS 204K CPY  001 A NOHOLD NO
KEEP
00: 0000003 FILES CHANGED
00: 0000003 FILES CHANGED
...
Kernel command line: ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc
                     TERM=dump HostIP=9.12.7.6 Hostname=virtcook6.itso.ibm.com
                     Gateway=9.12.4.1 Netmask=255.255.240.0 Layer2=1
                     ReadChannel=0.0.0600 WriteChannel=0.0.0601
                     DataChannel=0.0.0602 OSAHWAddr=
                     Nameserver=9.12.6.7 Portname= Portno=0
                     Install=ftp://9.12.7.8/SLES-11-SP3
                     UseVNC=1 VNCPassword=12345678
                     UseSSH=1 SSHPassword=12345678
                     InstNetDev=osa OsaInterface=qdio OsaMedium=eth Manual=1
...
>>> Linuxrc v3.3.91 (Kernel 3.0.76-0.9-default) <<<

Main Menu

1) Start Installation
2) Settings
3) Expert
4) Exit or Reboot
```

11. Select **1** for Start Installation:

```
Start Installation

1) Start Installation or Update
2) Boot Installed System
```

```
3) Start Rescue System
```

12. Select **3** for Start Rescue System. The rest of the startup procedure just follows the normal installation process. After bootup, you are presented a login prompt:

```
Master Resource Control: runlevel 3 has been   [80C [10D [1mreached [m

Rescue login:
```

13. Enter root as the login user.
14. You are asked about the terminal type to use. In doubt, use the value `vt100`. Only line mode can be used.
15. If a special key combination like [Ctrl]-[C] is needed in this terminal, write the characters ^C instead.
16. To leave the rescue system, type **exit**.

This section has described how to rescue a damaged Linux system.

26.2 Set up Memory Hotplugging

Linux *Memory Hotplug* allows the amount of memory in a Linux system to be increased or decreased without a reboot. You must first have standby memory defined to the virtual machine in which Linux is running. You can issue the **`CP DEFINE STORAGE`** command to configure standby memory (storage). Linux can then exploit the standby memory using the Service Call (**`SERVC`**) instruction.

To set up standby storage for Linux memory hotplug, using `LINUX1` as the virtual machine, perform the following steps:

1. Modify the `LINUX1` directory entry by adding a **`COMMAND`** statement. This will give the virtual machine an additional 768 MB of standby memory:

```
USER LINUX153 LNX4VM 256M 1G G
  INCLUDE LNXDFLT
  COMMAND DEFINE STORAGE 256M STANDBY 768M
  OPTION APPLMON
  MDISK 100 3390 3339 3338 UM63A9 MR LNX4VM LNX4VM LNX4VM
  MDISK 101 3390 6677 3338 UM63A9 MR LNX4VM LNX4VM LNX4VM
```

2. Bring the changes online with DirMaint, or with **`DIRECTXA`** if you are not using DirMaint.
3. Shut down the Linux system running on `LINUX1`. This can be done a number of ways, but because you are logged on to MAINT, it can be accomplished with the **`SIGNAL SHUTDOWN`** command:

```
==> signal shutdown linux1
```

4. Within about 30 seconds, you should see notification that the system went down cleanly and the virtual machine was logged off:

```
HCPSIG2113I User LINUX153 has reported successful termination
USER DSC   LOGOFF AS  LINUX153  USERS = 16    AFTER SIGNAL
```

5. Log on to `LINUX1`. You should see the standby memory reported:

```
LOGON LINUX1
00: z/VM Version 6 Release 2.0, Service Level 0000 (64-bit),
00: built on IBM Virtualization Technology
00: There is no logmsg data
00: FILES: 0001 RDR,   NO PRT,   NO PUN
00: LOGON AT 07:34:25 EDT SATURDAY 10/08/11
```

```
00: Command complete
00: NIC 0600 is created; devices 0600-0602 defined
00: NIC 0600 is connected to VSWITCH SYSTEM VSW1
00: Command complete
00: NIC 0700 is created; devices 0700-0702 defined
00: NIC 0700 is connected to VSWITCH SYSTEM VSW2
00: STORAGE = 256M MAX = 1G INC = 2M STANDBY = 768M RESERVED = 0
00: Storage cleared - system reset.
```

6. Answer **y**es to boot Linux:

```
DMSACP723I A (191) R/O
DMSACP723I C (592) R/O
DIAG swap disk defined at virtual address 300 (64989 4K pages of swap space)
DIAG swap disk defined at virtual address 301 (129981 4K pages of swap space)
Do you want to IPL Linux from minidisk 100? y/n
y
...
```

7. Start an SSH session as root and view the memory in the `/sys/` file system. Change directory to `/sys/devices/system/memory/` and list the files:

```
# cd /sys/devices/system/memory
# ls
block_size_bytes  memory0  memory1  memory2  memory3
```

8. Type the `block_size_bytes` file with the **`cat`** command:

```
# cat block_size_bytes
10000000
```

   This number is the number of bytes in hexadecimal. 10000000 in hex is 256 M in decimal. So the block size is 256 MB and there are four blocks: `memory0-memory3`, which are represented as directories. Each of the memory blocks has a state, which is represented as a file.

9. Show the state of each memory block with the following command:

```
# cat memory*/state
online
offline
offline
offline
```

   This shows that the first 256 MB is online and the next three blocks are offline.

10. You can also show information about memory with the **`free -m`** command:

```
# free -m
             total       used       free     shared    buffers     cached
Mem:           241        165         75          0         18         54
-/+ buffers/cache:         92        148
Swap:          761          0        761
```

   This shows 241 MB of free memory available (some of the memory is used internally by Linux).

11. You can turn on memory by sending the string **online** to the state file. Turn on an additional 512 MB of memory with the following commands:

```
# echo online > memory1/state
# echo online > memory2/state
```

12. Show that the memory is now online:

```
# cat memory*/state
online
online
```

```
online
offline
```

13. Again, confirm with the **free -m** command:

```
# free -m
total       used       free     shared    buffers     cached
Mem:           752        167        584          0         11         60
-/+ buffers/cache:         96        655
Swap:         1273          0       1273
```

This shows that 752 MB of free memory is now available - 511 MB more than before.

14. You can also give the memory back by echoing **offline** to the state file:

```
# echo offline  > memory1/state
# echo offline  > memory2/state
```

15. Verify that the memory has to be returned:

```
# cat memory*/state
online
offline
offline
offline
# free -m
total       used       free     shared    buffers     cached
Mem:           240        163         76          0         11         60
-/+ buffers/cache:         92        147
Swap:         1273          0       1273
```

This section has shown how to configure virtual machines with standby memory and how to "hot-plug" the memory from Linux. This function can increase your system's performance and availability.

26.3 Utilize the cpuplugd service

The **cpuplugd** service allows Linux to enable or disable CPUs and memory, based on a set of rules. It can improve performance by setting the correct number of processors and amount of memory for Linux systems depending on their current load. It can also prevent the Linux scheduler from queue balancing in partial load situations.

More information about **cpuplugd** can be found in the manual *Linux on System z Device Drivers, Features and Commands* for RHEL and SLES, on the web at the following sites:

- http://www.ibm.com/developerworks/linux/linux390/documentation_red_hat.html
- http://www.ibm.com/developerworks/linux/linux390/documentation_novell_suse.html

26.3.1 Determine the virtual CPUs being used

To start work with **cpuplugd**, perform the following steps:

1. Start an SSH session to Linux and determine how many CPUs that Linux has online. Write a short bash script, **lscpus**, to save typing:

```
# cd /usr/local/sbin
# vi lscpus
#!/bin/bash
# script to list the number and status of virtual CPUs
for i in /sys/devices/system/cpu/cpu*
do
```

```
  echo $i
  cat $i/online
done
```

2. Save the file and the set it to be executable:

```
# chmod +x lscpus
```

3. Observe the status of the **cpuplugd** service:

```
# service cpuplugd status
cpuplugd (pid  1574) is running...
```

This shows that **cpuplugd** is running

4. Wait a few minutes and run the **lscpus** script again:

```
# lscpus
/sys/devices/system/cpu/cpu0
1
/sys/devices/system/cpu/cpu1
0
/sys/devices/system/cpu/cpu2
0
/sys/devices/system/cpu/cpu3
0
/sys/devices/system/cpu/cpu4
0
/sys/devices/system/cpu/cpu5
0
/sys/devices/system/cpu/cpu6
0
/sys/devices/system/cpu/cpu7
0
/sys/devices/system/cpu/cpu8
0
/sys/devices/system/cpu/cpu9
0
```

The output shows that now only one of the 10 virtual CPUs are active. The **cpuplugd** service turned off the other nine.

5. The **cpuplugd** configuration file is `/etc/sysconfig/cpuplugd`. Some middleware products recommend a minimum of two virtual processors. If most of your Linux servers will be running a workload that recommends two processors, change the default for `CPU_MIN` to 2. An exception would be when only a single physical processor is available. View the non-comments and lines that are not blank in the configuration file with the following command:

```
# cd /etc/sysconfig
# egrep -v '^$|^#' cpuplugd
CPU_MIN="1"
CPU_MAX="0"
UPDATE="10"
CMM_MIN="0"
CMM_MAX="8192"
CMM_INC="256"
HOTPLUG="(loadavg > onumcpus + 0.75) & (idle < 10.0)"
HOTUNPLUG="(loadavg < onumcpus - 0.25) | (idle > 50)"
MEMPLUG="0"
MEMUNPLUG="0"
```

The default rules for the plugging and unplugging of CPUs in the configuration file is as follows:

```
HOTPLUG = "(loadavg > onumcpus +0.75) & (idle < 10.0)"
```

```
HOTUNPLUG = "(loadavg < onumcpus -0.25) | (idle > 50)"
```

Where the variables in the statements have the following meaning:

`loadavg`	The current average CPU load
`onumcpus`	The number of CPUs that are online
`runable_proc`	The current number of processes that can be run
`idle`	The current idle percentage

These CPU hot plugging and unplugging values will be used in the next section. In the default setup, **cpuplugd** will only make changes to the virtual processor configuration. The auto adaptive adjustment of the memory using the `cmm` feature (module) is deactivated by default and also not available when running in a native LPAR environment.

26.3.2 Generating a workload to see cpuplugd work

You can now generate a workload to show how the **cpuplugd** will turn on CPUs.

> **Important:** Running the following command will generate significant CPU use. Verify that there is not a mission-critical workload running on this z/VM LPAR because this test may affect it. Also, be sure to kill the processes after seeing **cpuplugd** in action.

Perform the following steps:

1. Put 10 looping jobs in the background with the following **for** loop:

```
# for i in `seq 1 10`
> do
>   bash -c "cat /dev/zero > /dev/null" &
> done
[1] 2441
[2] 2442
[3] 2443
[4] 2444
[5] 2445
[6] 2446
[7] 2447
[8] 2448
[9] 2449
[10] 2453
```

2. See that the jobs are running (you can also use the **top** command):

```
# pstree -G | grep cat
    +-sshd---sshd---bash---10*[bash---cat]
```

3. Now run **lscpus** every so often. The following example shows that, after a minute or so, **cpuplugd** has started five of the nine spare processors:

```
# lscpus
/sys/devices/system/cpu/cpu0
1
/sys/devices/system/cpu/cpu1
1
/sys/devices/system/cpu/cpu2
1
/sys/devices/system/cpu/cpu3
1
/sys/devices/system/cpu/cpu4
1
/sys/devices/system/cpu/cpu5
```

```
1
/sys/devices/system/cpu/cpu6
0
/sys/devices/system/cpu/cpu7
0
/sys/devices/system/cpu/cpu8
0
/sys/devices/system/cpu/cpu9
0
```

After a few more minutes, all of the CPUs should be activated.

4. Kill the processes with the **killall** command, then verify that the loops have stopped:

```
# killall cat
bash: line 1:  2450 Terminated              cat /dev/zero > /dev/null
bash: line 1:  2452 Terminated              cat /dev/zero > /dev/null
bash: line 1:  2451 Terminated              cat /dev/zero > /dev/null
bash: line 1:  2457 Terminated              cat /dev/zero > /dev/null
bash: line 1:  2456 Terminated              cat /dev/zero > /dev/null
[1]   Exit 143                bash -c "cat /dev/zero > /dev/null"
[2]   Exit 143                bash -c "cat /dev/zero > /dev/null"
...
# pstree -G | grep cat
```

No output shows that the processes to create a workload have been stopped.

26.3.3 Setting memory sizes with cpuplugd

Memory sizes can also be set by the **cpuplugd** service. However, unlike CPUs, there is no good generic default value. The following example is in the *Device Drivers* book:

```
MEMPLUG = "swaprate > freemem+10 & freemem+10 < apcr"
MEMUNPLUG = "swaprate < freemem + 10000"
```

However, this is just a starting point to explain the syntactical structure of a rule. Do not use this configuration in production. You should test any setting that you want to implement against a representative workload that your Linux systems will be running. Details are beyond the scope of this section.

26.4 Hardware cryptographic support for OpenSSH

This section shows how to copy a test file with OpenSSH, first without any crypto acceleration. Then, crypto acceleration for OpenSSH is enabled and the same file is copied again. A much higher throughput rate should be observed. The prerequisite for using hardware cryptography is to have a firmware level of LIC 3863 installed on your System z CEC.

This section is based on the white paper *First experiences with hardware cryptographic support for OpenSSH with Linux for System z*, by Manfred Gnirss, Winfried Münch, Klaus Werner, and Arthur Winterling, on the web at the following site:

http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101690

This section shows only a single example of crypto acceleration. For a much more complete and detailed analysis, see the white paper.

To test copying a file with and without cryptographic acceleration, perform the following steps:

1. Start an SSH session as root to any Linux.
2. Create a 200 MB test file for copying in the /tmp/ directory:

```
# cd /var
# dd if=/dev/zero of=testdata.txt bs=1048576 count=200
200+0 records in
200+0 records out
209715200 bytes (210 MB) copied, 17.87 s, 11.7 MB/s
# ls -lh testdata.txt
-rw-r--r-- 1 root root 200M Dec 28 07:57 testdata.txt
```

3. Copy the file locally with the **scp** command, two times with specific encryption algorithms and once without, prefixing all with the **time** command:

```
# time scp -c 3des-cbc testdata.txt localhost:/dev/null
testdata.txt                                  100%  200MB   6.3MB/s   00:32

real    0m31.598s
user    0m15.499s
sys     0m0.375s

# time scp -c aes128-cbc testdata.txt localhost:/dev/null
testdata.txt                                  100%  200MB  18.2MB/s   00:11

real    0m11.324s
user    0m5.006s
sys     0m0.367s

# time scp testdata.txt localhost:/dev/null
testdata.txt                                   100%  200MB  18.2MB/s   00:11

real    0m11.766s
user    0m5.468s
sys     0m0.368s
```

   The output shows throughputs of about 6.3, 18.2, and 18.2 MBps and user times of about 15.5, 5.0, and 5.47 seconds.

4. Determine if the necessary cryptographic-related RPMs are installed:

```
# rpm -qa | grep openssl-ibmca
```

   No output shows that they are not installed.

Red Hat Enterprise Linux

If you are using RHEL 6.4, perform the following steps:

1. Install the RPMs with the **yum install** command:

```
# yum -y install openssl-ibmca openssl-ibmca.s390
...
Installed:
  openssl-ibmca.s390 0:1.2.0-2.el6_2.1

Dependency Installed:
  keyutils-libs.s390 0:1.4-4.el6        krb5-libs.s390 0:1.10.3-10.el6
  libcom_err.s390 0:1.41.12-14.el6      libselinux.s390 0:2.0.94-5.3.el6
  openssl.s390 0:1.0.0-27.el6           zlib.s390 0:1.2.3-29.el6

Complete!
```

2. Verify that the RPMs are now installed:

```
# rpm -qa | egrep "libica|ibmca"
```

```
libica-2.1.0-3.el6.s390x
openssl-ibmca-1.2.0-2.el6_2.1.s390x
openssl-ibmca-1.2.0-2.el6_2.1.s390
```

3. Verify that CP Assist for Cryptographic Function (CPACF) operations are supported:

```
# icainfo
The following CP Assist for Cryptographic Function (CPACF) operations are
supported by libica on this system:

SHA-1:        yes
SHA-256:      yes
SHA-512:      yes
DES:          yes
TDES-128:     yes
TDES-192:     yes
AES-128:      yes
AES-192:      yes
AES-256:      yes
PRNG:         yes
CCM-AES-128:  yes
CMAC-AES-128: yes
CMAC-AES-192: yes
CMAC-AES-256: yes
```

4. Make a backup of the SSL configuration file, `/etc/ssl/openssl.cnf`:

```
# cd /etc/pki/tls
# cp openssl.cnf openssl.cnf.orig
```

5. Append the sample SSL configuration file in the package documentation to the actual SSL configuration file, `/etc/openssl.cnf`:

```
# cat /usr/share/doc/openssl-ibmca-1.2.0/openssl.cnf.sample-s390x >> openssl.cnf
```

6. Make a symbolic link to the file `/usr/lib64/openssl/engines/libibmca.so`:

```
# cd /usr/lib64
# ln -s openssl/engines/libibmca.so
# ls -l libibmca.so
lrwxrwxrwx. 1 root root 27 Oct 20 16:47 libibmca.so -> openssl/engines/libibmca.so
```

You should now have the cryptographic packages installed on Red Hat.

SUSE Enterprise Linux Server

If you are using SLES 11 SP3, perform the following steps:

1. Install the RPMs with the **`zypper install`** command:

```
# zypper in openssl-ibmca openssl-ibmca-32bit libica-2_1_0 libica-2_1_0-32bit
Loading repository data...
Reading installed packages...
Resolving package dependencies...

The following NEW packages are going to be installed:
  glibc-32bit libica-2_1_0 libica-2_1_0-32bit libopenssl0_9_8-32bit openssl-ibmca
openssl-ibmca-32bit zlib-32bit

7 new packages to install.
Overall download size: 2.3 MiB. After the operation, additional 6.6 MiB will be used.
Continue? [y/n/?] (y):
...
Installing: openssl-ibmca-1.2.0-141.13.1 [done]
Installing: zlib-32bit-1.2.7-0.10.128 [done]
```

```
Installing: libopenssl0_9_8-32bit-0.9.8j-0.50.1 [done]
Installing: openssl-ibmca-32bit-1.2.0-141.13.1 [done]
Installing: libica-2_1_0-32bit-2.1.0-0.12.28 [done]
```

2. Show the new RPMs with the following command:

```
# rpm -qa | egrep "libica|ibmca"
openssl-ibmca-1.2.0-141.13.1
libica-2_1_0-2.1.0-0.12.28
libica-2_1_0-2.1.0-0.12.28
openssl-ibmca-32bit-1.2.0-141.13.1
```

3. Verify that CP Assist for Cryptographic Function (CPACF) operations are supported:

```
# icainfo
The following CP Assist for Cryptographic Function (CPACF) operations are
supported by libica on this system:
# icainfo
The following CP Assist for Cryptographic Function (CPACF) operations are
supported by libica on this system:
SHA-1:        yes
SHA-256:      yes
SHA-512:      yes
DES:          yes
TDES-128:     yes
TDES-192:     yes
AES-128:      yes
AES-192:      yes
AES-256:      yes
PRNG:         yes
CCM-AES-128:  no
CMAC-AES-128: no
CMAC-AES-192: no
CMAC-AES-256: no
```

You should now have the cryptographic packages installed on SUSE.

Red Hat Enterprise Linux or SUSE Linux Enterprise Server

To configure cryptography on either Linux distribution, perform the following steps.

1. Make a backup of the SSL configuration file, `/etc/ssl/openssl.cnf`:

```
# cd /etc/ssl
# cp openssl.cnf openssl.cnf.orig
```

2. Append the sample SSL configuration file in the package documentation to the actual SSL configuration file, `/etc/openssl.cnf`:

```
# cat /usr/share/doc/packages/openssl-ibmca/openssl.cnf.sample >> openssl.cnf
```

3. Edit the appended file and search for the line with the **`openssl_conf`** variable. Move that line from the bottom to the top and save the file, as shown in the following example:

```
# vi openssl.cnf
/openssl_conf
#
# OpenSSL example configuration file.
# This is mostly being used for generation of certificate requests.
#

# This definition stops the following lines choking if HOME isn't
# defined.
HOME                    = .
RANDFILE                = $ENV::HOME/.rnd
```

```
openssl_conf = openssl_def
...
```

4. Rerun the same **scp** commands:

```
# cd /var
# time scp -c 3des-cbc testdata.txt localhost:/dev/null
testdata.txt                                        100%  200MB  66.7MB/s   00:03

real    0m3.419s
user    0m1.362s
sys     0m0.358s
# time scp -c aes128-cbc testdata.txt localhost:/dev/null
testdata.txt                                        100%  200MB 100.0MB/s   00:02

real    0m2.410s
user    0m0.890s
sys     0m0.355s
```

5. Delete the test file:

```
# rm /tmp/testdata.txt
```

6. To set a preferred cipher for the connections, edit the `/etc/ssh/sshd_config` file and add the following line at the end of the configuration file:

```
# cd /etc/ssh
# vi sshd_config
...
Ciphers aes128-cbc
```

7. The changed default for the cipher improves the throughput:

```
# time scp testdata.txt localhost:/dev/null
testdata.txt                              100%  200MB 100.0MB/s   00:02

real    0m2.423s
user    0m0.893s
sys     0m0.357s
```

You should see an improved throughput as a result of using the cryptographic hardware.

26.5 The X Window System

For many years, UNIX like operating systems have been using the X Window System (commonly just “X”). This system was designed to provide client/server, hardware-independent, and network-enabled graphical environment. The current version is X11, which is widely used on UNIX and Linux platforms.

Confusion often arises among new X users regarding the concept of client and server because client and server are defined from an application point of view while other protocols such as SSH, Telnet, and FTP are defined from an end-user point of view. In X, the server runs on the hardware with the mouse, keyboard, and monitor (usually a workstation or a desktop), while the client runs on the UNIX or Linux server. Many Linux desktop users do not recognize this difference because they often run both the server and client on their desktop.

It is a common practice to connect from a PC (SSH client) to remote Linux (SSH server) and then run an X application. It runs on remote Linux (X client) and displays on local PC (X server).

The X communication protocol by its nature is not secure at all. For this reason it is often used together with SSH protocol, which tunnels X11 traffic using encrypted (and thus secure) communications.

X11 itself provides the ability to display graphics on raster display, nothing more. If the user wants to be able to move, resize, and otherwise manage windows, a *window manager* is needed. There are many window managers available; some are lightweight while some are more robust. So using a window manager is a good idea because it provides functionality, which one expects from a graphical user interface (GUI).

When you have Linux installed on your workstation, a window manager is probably not enough. Here you want a full desktop environment with menus, icons, taskbars, and so on, such as Gnome and KDE. Installing GNOME or KDE on System z is discouraged because they are resource-intensive. Installing The X Window System is also not recommended.

26.5.1 VNC Server

As mentioned earlier, the X server is run where the mouse, keyboard, and monitor are located - on the workstation. In a nutshell, VNC Server provides a virtual workstation with all these peripherals (virtual). The VNC server starts an embedded X server. Then, any X-based application can send its output to this X server, regardless of if the applications are local or remote to the X server.

To interact with the X server, one uses VNC client on a workstation, as described in section 3.2, “Set up a VNC client” on page 41. The VNC server customization is described in section 9.2.4, “Configure the VNC server” on page 186. In our experience, this is all that you need if you want to run X applications from time to time.

One big advantage of VNC is that it is session-oriented. If communication to VNC server is lost, a new connection is reestablished to the session as it was. Also, applications in a disconnected VNC session still continue to run.

26.5.2 X server on a workstation

If for some reason VNC is not acceptable, it is possible to use a standard X server on a workstation. Since Linux users usually know the X Window System, an X server running on Windows is described in this section.

There are many commercial and free X Window servers available for Windows. In the following examples, XliveCD is used, which provides a free X server that is based on Cygwin. It can be run directly from a CD without requiring an installation. See the following website:

`http://xlivecd.indiana.edu`

Any X application will send its output to an address defined with the **`-display`** parameter or, if not provided, to an address specified in the `DISPLAY` environment variable. Following is an example that uses the **`xclock`** command (you might have to first install it with the **`yum -y install xclock`** command):

```
gpok224:~ # xclock
Error: Can't open display:
```

There is no display specified for the **`xclock`** command and it will terminate.

Display is specified by setting the DISPLAY environment variable.

```
gpok224:~ # export DISPLAY=9.145.177.158:0
```

```
gpok224:~ # xclock
No protocol specified
Error: Can't open display: 9.145.177.158:0
```

This command failed because the XliveCD requires an explicit command to allow remote hosts to connect to it. When the command **xhost +** (plus means to add authorized hosts) is run in XliveCD command line, **xclock** can finally display on Windows. Remember the program itself runs on a remote Linux.

```
gpok224:~ # xclock &
[1] 21915
```

The **xhost +** command allows any host to access the X Server. From a security point of view, this might not be a good idea. A better practice would be to allow only specific hosts to access X server.

There is still another security concern because X11 protocol itself is not secure. Using SSH tunneling removes this security exposure. SSH tunneling also prevents firewalls and NAT from breaking X11 communications.

It is possible to use an external SSH client, which allows X11 forwarding, or an SSH client embedded in XliveCD itself. Both options are shown.

Using PuTTY

To use PuTTY for X11 forwarding, select X11 forwarding as shown in Figure 26-1.

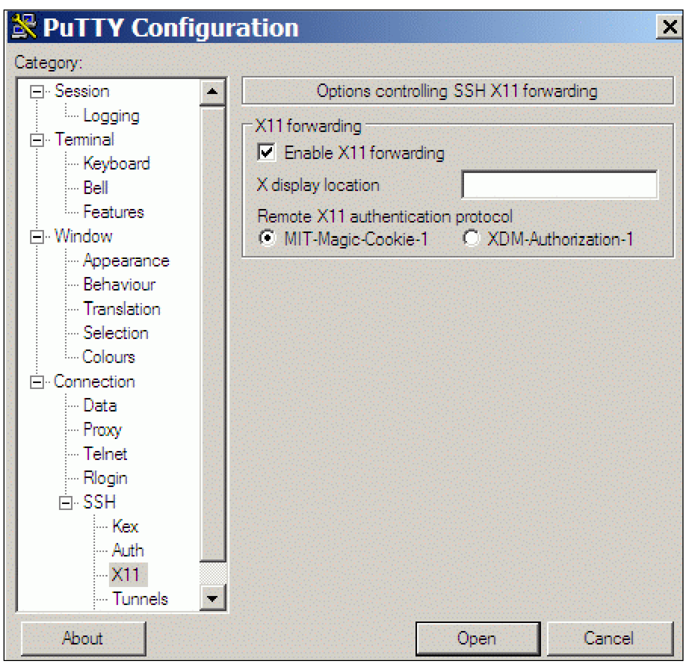


Figure 26-1 Allow X11 forwarding in PuTTY

When connected to a remote Linux system with X11 forwarding enabled, the `DISPLAY` environment variable contains the special value of **localhost:10.0**, which tells PuTTY to forward X11 protocol over SSH to SSH client address. In this case, there is no need to enter the **xhost** command because the connection appears to X server as a local one.

Using embedded SSH

It is also possible to achieve X11 forwarding with an embedded SSH client with an -X parameter. Again, no **xhost** command is needed.

There are many ways how to achieve the same results. It is up to you to choose a solution that suits the purpose best.

26.6 Centralizing home directories for LDAP users

In previous versions of this book, there was a section on a traveling `/home/` directory using LDAP, NFS, and automount. In the interest of time, this section has been removed. See section 13.3 in the IBM Redbooks publication *z/VM and Linux on IBM System z The Virtualization Cookbook for Red Hat Linux Enterprise Server 5.2*, SG24-7492, on the web at the following site:

http://www.redbooks.ibm.com/abstracts/sg247492.html

26.7 Setting up the Linux Terminal Server

A Linux Terminal Server on System z allows access to the console without a functioning TCP/IP stack on Linux. Further, it has an interface that allows for character-addressable functions needed by tools such as **vi**.

The official documentation for setting up Linux Terminal Server is on the web at the following site:

http://public.dhe.ibm.com/software/dw/linux390/docu/l26dht00.pdf

That document has many different options on how Linux Terminal Server can be set up, while this section supplies just one step to get it running by using the **iucvtty** command.

To set up the Linux Terminal Server, some work has to be done on z/VM and some on Linux. The work done on Linux will differ depending on whether the system is Red Hat or SUSE. To set up Linux Terminal Server, perform some or all of the following steps depending on the Linux distribution you are working with:

- "z/VM configuration for Linux Terminal Server" on page 500
- "Red Hat Enterprise Linux 6.4 configuration for Linux Terminal Server" on page 501
- "SLES 11 SP3 configuration for Linux Terminal Server" on page 502

26.7.1 z/VM configuration for Linux Terminal Server

To configure z/VM for Linux Terminal Server, perform the following steps:

1. You should have the line `IUCV ALLOW` in the user directory profile `LNXDFLT` if you followed the steps in section 5.11, "Create identity LNXADMIN for Linux administration" on page 96:

```
PROFILE LNXDFLT
  COMMAND SET VSWITCH VSW1 GRANT &USERID
  COMMAND DEFINE NIC 600 TYPE QDIO
  COMMAND COUPLE 600 TO SYSTEM VSW1
  COMMAND SET VSWITCH VSW2 GRANT &USERID
  COMMAND DEFINE NIC 700 TYPE QDIO
  COMMAND COUPLE 700 TO SYSTEM VSW2
```

```
  CPU 00 BASE
  CPU 01
  IPL CMS
  MACHINE ESA 8
  IUCV ALLOW
...
```

Having this line will allow all Linux virtual machines to connect to the Terminal Server using IUCV.

2. Add two lines to the bottom of the user directory entry for the Terminal Server virtual machine. This example uses `LNXADMIN` and shows how to update the `IDENTITY` definition with DirMaint. Note the file number that is sent to the reader (***105*** in this example):

```
==> dirm for lnxadmin get
DVHXMT1191I Your GET request has been sent for processing to DIRMAINT at
DVHXMT1191I ZVM63A.
DVHREQ2288I Your GET request for LNXADMIN at * has been accepted.
 DVHGET3304I Directory entry  LNXADMIN is now locked.
RDR FILE 0105 SENT FROM DIRMAINT PUN WAS 0149 RECS 0011 CPY  001 A NOHOLD NOKEEP
 DVHREQ2289I Your GET request for LNXADMIN at * has completed; with RC
 DVHREQ2289I = 0.
```

3. Receive the file from the reader, `105`, in this example:

```
==> receive 105
File LNXADMIN DIRECT A0 created from LNXADMIN DIRECT A0 received from DIRMAINT a
t ZVM63A
```

4. Edit the `LNXADMIN DIRECT` file and add two lines near the bottom:

```
==> x lnxadmin direct
IDENTITY LNXADMIN LNX4VM 512M 1G BDEG
 INCLUDE LNXDFLT
 BUILD ON ZVM63A USING SUBCONFIG LNXADM-1
 BUILD ON ZVM63B USING SUBCONFIG LNXADM-2
 IUCV ANY
 OPTION MAXCONN 128
 OPTION LNKNOPAS
```

These two lines will allow IUCV communication and up to 128 concurrent terminal sessions.

5. Replace the modified directory entry with the following **DIRM REPLACE** command:

```
==> dirm for lnxadmin rep
...
 DVHREQ2289I Your REPLACE request for LNXADMIN at * has completed; with DVHREQ2289I RC =
0.
```

z/VM should now be configured for the Linux Terminal Server.

26.7.2 Red Hat Enterprise Linux 6.4 configuration for Linux Terminal Server

To configure Linux Terminal Server on RHEL 6.4, perform the following steps:

1. On target Linux systems, add one line at the bottom of the file `/etc/inittab`:

```
# cd /etc
# vi inittab
...
t1:2345:respawn:/usr/bin/iucvtty hvc0
```

2. On target Linux systems, add one entry to the file `/etc/zipl.conf`:

```
# vi zipl.conf
```

```
...
parameters = "root=/dev/disk/by-path/ccw-0.0.0100-part1 vmpoff=LOGOFF vmhalt=LOGOFF
hvc_iucv=8 console=hvc0 TERM=dumb resume=/dev/disk/by-path/ccw-0.0.0301-part1"
...
```

3. Run zipl:

```
# zipl
...
```

4. To test the changes, start a session from the Terminal Server to a Linux virtual machine, ZNTC75 in this example:

```
# iucvconn zntc75 lnxhvc0
```

26.7.3 SLES 11 SP3 configuration for Linux Terminal Server

On SLES 11 SP3, the IUCV HVC consoles are available by default from the /etc/inittab file. Therefore, it is only necessary to configure the terminal server but no changes are required on the systems that need to be connected to.

To configure Linux Terminal Server on SLES 11 SP3, perform the following steps:

1. Start an SSH session as root to LNXADMIN.
2. Run **yast** → Network Services → IUCV Terminal Server.
3. Select z/VM IDs then press Enter and Tab to switch to the z/VM IDs input field.
4. Add all the machines that LNXADMIN should connect to.
5. Click the **IUCVConn** tab.
6. Select **Enable IUCVConn on Login**.
7. Enter a password for the IUCVCONN user and confirm it.
8. Click **OK** to finish the configuration.
9. Start a new SSH session to lnxadmin. Use one of the machines that were configured and is running Linux at that moment. The password for the connection is just the iucvconn password that was entered before:

```
# ssh linux6@virtcook8.itso.ibm.com
login as: linux6
Using keyboard-interactive authentication.
Password:
iucvconn_on_login: Connecting to linux6 (terminal ID: lnxhvc0)

Welcome to SUSE Linux Enterprise Server 11 SP3  (s390x) - Kernel 3.0.76-0.9-default
(hvc0).

virtcook6 login:
```

10. You are now connected to the IUCV terminal hvc0 of the z/VM guest LINUX6 and are prompted for a login. Note, that no network connection to that guest is needed at this time.
11. To terminate the connection, a special key sequence is required. The default for this is [Ctrl]-[SHIFT]-[-] [.]. (for the editor: in words this is the key combination control underscore followed by a dot).

26.8 Redefine command privilege classes

While a general Linux guest privilege class of G is enough, there are situations when some virtual machines need access to special commands. An example would be LNXADMIN user that needs to access the **FLASHCOPY** command, which requires a B class user privilege. For this reason, LNXADMIN has more access than is really needed as it has an access to all B class commands. One possible solution is to create a new user class, which will have access to the **FLASHCOPY** command only and not to other B class commands. To do this, perform the following steps:

1. Modify the **FLASHCOPY** command to be available to a new class Z:

   ```
   ==> cp modify cmd flashcopy privclass z
   ```

2. To make this change permanent, modify the SYSTEM CONFIG file with the **MODIFY CMD** statement:

   ```
   modify cmd flashcopy privclass z
   ```

3. Modify the user directory record for LNXADMIN:

   ```
   IDENTITY LNXADMIN LNX4VM 256M 1G GZ
   ...
   ```

4. After the modified user directory is online and LNXADMIN logs on, check the new privilege classes:

   ```
   ==> q privclass
   Privilege classes for user LNXADMIN
           Currently: GZ
           Directory: GZ
   ```

   Now LNXADMIN can execute G class commands as well as the **FLASHCOPY** command.

26.9 Use Crypto Express to seed /dev/random

Linux has a difficult time getting enough entropy from low interrupts and keyboard events on mainframe computers just as on newer distributed servers. Largely, this comes from the fact that the upper interrupts are not used to seed the entropy pool, and usually there is no keyboard or mouse attached to the system. With SUSE Linux Enterprise Server, there is an entropy generator installed by default that increases entropy when needed.

To check the availability of entropy on a server, do the following:

1. Log on to the server:

   ```
   # ssh root@virtcook8
   ```

2. Check the current entropy:

   ```
   # cat /proc/sys/kernel/random/entropy_avail
   3965
   ```

The values of the available entropy are always kept 128 - 4096. Therefore, 3965 is a good value. However, although **haveged** is likely to produce good entropy, some environments might want to go with the hardware random generator from the Crypto Express card. To check what happens without entropy generator, do the following steps:

1. Start an SSH session as root to a Linux system.
2. Disable the **haveged** service:

   ```
   # rchaveged stop
   Shutting down haveged daemon                                              done
   ```

3. Run a small loop to see how the entropy develops over time:

```
# for i in {1..20}; do cat /proc/sys/kernel/random/entropy_avail ; done
2985
2857
...
681
553
```

   The numbers getting smaller can be explained with the page randomization of the Linux kernel. For each new process, the Linux kernel uses a small amount of entropy, which leads to a decrease of the available entropy. As entropy decrease, programs like **sshd** and web servers with SSL will have issues.

To enable the hardware random number generator from the Crypto Express card, a dedicated crypto domain is needed. To create this, perform the following steps:

1. Log as MAINT.
2. To check for an available crypto domain, do the following:

```
==> query crypto ap
AP 03 CEX4C  Domain 03 available    shared                unspecified
```

3. Dedicate a crypto domain to one Linux guest. Edit the user directory entry and add the following statement:

```
CRYPTO DOMAIN 3  APDED 3
```

4. Bring the changes to the user directory online.
5. Query the crypto device again:

```
==> query crypto ap
...
```

6. Log on to the virtual machine and IPL Linux.
7. Start an SSH session as root.
8. Turn the **haveged** service off:

```
# chkconfig haveged off
# rchaveged stop
```

9. Make sure that the package **libica-2_1_0** is installed:
10. # **zypper in libica-2_1_0**
11. Enable the service **z90crypt**:
12. # **chkconfig z90crypt on**
13. # **rcz90crypt start**
14. Check the availability of entropy with an endless loop:
15. # **watch -n 0.1 cat /proc/sys/kernel/random/entropy_avail**
16. Stop the program by issuing [CTRL]-C.

This section has shown how to utilize the cryptographic hardware to increase entropy to generate numerous random bytes.

Part 5

Appendixes

This section consists of the following appendixes:

A

References and cheat sheets

This appendix refers to additional material that can be downloaded from the Internet as described below.

Related books

The following publications can be used as information sources:

- *Documentation for System z Linux Development stream*: Available on the web at the following sites:
  - http://www.ibm.com/developerworks/linux/linux390/documentation_red_hat.html
  - http://www.ibm.com/developerworks/linux/linux390/documentation_novell_suse.html
- *RHEL 6: IBM System z Architecture - Installation and Booting*:

  http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/6/html/Installation_Guide/pt-install-info-s390.html
- z/VM documentation. Start at the following website:

  http://www.vm.ibm.com/library
- From the preceding website, see the **z/VM bookshelf** line near the top and click **V6.3**. The following books are especially useful:
  - z/VM CP Messages and Codes
  - z/VM TCP/IP Messages and Codes
  - The Program Directory for Performance Toolkit for VM
  - z/VM CP Commands and Utilities Reference
  - z/VM CP Planning and Administration
  - z/VM Getting Started with Linux on System z
  - z/VM TCP/IP Planning and Customization
  - *z/VM Performance Toolkit Guide*, SC24-6156-00
  - *z/VM Performance Toolkit Reference*, SC24-6157-00

- Redbooks publications. Start at the following website, then search for the listed publications:

  http://www.redbooks.ibm.com
  - *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

Online resources

These websites and URLs are also relevant as further information sources:

- The Linux for zSeries and S/390 portal:

  http://linuxvm.org
- The `IBMVM` list server:

  http://listserv.uark.edu/archives/ibmvm.html
- The `linux-390` list server:

  http://www2.marist.edu/htbin/wlvindex?linux-390
- SUSE Linux Enterprise Server 9 evaluation:

  http://www.novell.com/products/linuxenterpriseserver/eval.html
- z/VM publications:

  http://www.vm.ibm.com/pubs
- z/VM performance tips:

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

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 26-1 provides a summary and the location of important z/VM configuration files.

Table 26-1 Important z/VM configuration files

File	Location	Description
`SYSTEM CONFIG`	`PMAINT CF0`	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 virtual machines that are known to the system are defined here (assuming that a directory maintenance product is not being used).
`PROFILE 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.

File	Location	Description
`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. It is analogous to the `/etc/inittab` file in Linux.

Cheat sheets

This section contains quick references or “cheat sheets” for the XEDIT and vi editors.

XEDIT cheat sheet

`XEDIT` has line commands that are typed on the command line (===>) and prefix commands, which are typed over the line numbers on the left side of the window.

Line commands

```
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 occurrences of ‘string’ and hide other lines
bottom                     Move to the bottom of the file
top                        Move to the top of the file
down <n>                   Move down ‘n’ lines
up <n>                     Move up ‘n’ lines
file                       Save the current file and exit XEDIT
ffile                      Save the current file and exit but 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 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 where you can type text into the file. When you are in this mode, you see the text `--INSERT--` in the last line.
2. Command mode: “Esc” gets you out of input mode and into command mode. You can issue the following commands:

```
i              brings you back to input mode
dd             deletes a line and puts it in the buffer
<n>dd          delete <n> lines
x              delete a character
dw             delete a word
p              add the buffer past the current location
P              add the buffer before the current location
o              add a line and go into insert mode
/string        search for string
n              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 at the bottom of the window. You can issue the following commands:

```
: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
```

DirMaint cheat sheet

```
Add            Add a new user or profile directory entry
AMDisk         Adds a new minidisk
DEDicate       Add or delete an existing dedicate statements
DMDisk         Removes a minidisk
FILE           Add or replace a DirMaint control file
```

```
RLDCode         Reload DirMaint resident operating procedures
RLDExtn         Reload DirMaint CONFIG* DATADVH file
REView          Review a user or profile directory entry
MDisk           Change the access mode and passwords for minidisks
STorage         Change logon storage size
SEND            Request a copy of a DirMaint control file
SETOptn         Add, change, or delete CP options
CLAss           Change the CP class for a directory entry
SPEcial         Add or delete an existing special statement
```

B

Additional material

This book refers to additional material that can be downloaded from the Internet.

Locating the web material

The web material associated with this book is available on the Internet. You can find this at the following URL:

http://www.vm.ibm.com/devpages/mikemac/SG248147.tgz

Using the web material

The files associated with this book are in a *GNU* compressed tar file. To use the files associated with this book, see section 4.3, “Set up an NFS server” on page 49.

The additional web materials that accompany this book are in the following file:

File name	*Description*
SG248147.tgz	Code samples in compressed tar format

Within the tar file, the directory `SG248147/` contains the following subdirectories and files:

`disclaimer.txt`	Legal disclaimer
`README.txt`	Description file
`rhel64/`	Directory with files for RHEL 6.4
`rhel64/clone-1.0-11.s390x.rpm`	RHEL 6.4 clone RPM
`sles11sp3/`	Directory with files for SLES 11 SP3
`sles11sp3/clone.sh`	SLES 11 SP3 clone script
`sles11sp3/linux5.xml`	AutoYaST profile
`sles11sp3/boot.clone`	Init script for new clones
`sles11sp3/jeos.tgz`	Files associated with kiwi
`vm/`	Directory with files for z/VM
`vm/lnxmaint/`	Directory with files for LNXMAINT 192
`vm/lnxmaint/rhel64.exec`	EXEC to start an RHEL 6.4 installation

`vm/lnxmaint/sample.parm-rh6`	Sample RHEL 6.4 parameter file
`vm/lnxmaint/sample.conf-rh6`	Sample RHEL 6.4 configuration file
`vm/lnxmaint/sample.parm-s11`	Sample SLES 11 SP3 parameter file
`vm/lnxmaint/profile.exec`	Sample PROFILE EXEC for Linux IDs
`vm/lnxmaint/sles11s3.exec`	EXEC to start an SLES 11 SP3 installation
`vm/lnxmaint/swapgen.exec`	EXEC to define VDisk swap spaces
`vm/maint/`	Directory with files for MAINT 191
`vm/maint/callsm1.exec`	EXEC to test SMAPI
`vm/maint/cpformat.exec`	EXEC to format multiple DASD volumes
`vm/maint/ssicmd.exec`	EXEC to run a command on all SSI members

System requirements for downloading the web material

The web material requires the following system configuration:

Hard disk space: 41 KB
Operating System: Linux

Downloading and extracting the web material

See section 4.2.1, “Download files associated with this book” on page 47 for details about how to use these files.

This section lists source code associated with this book. The following sections are included:

- “z/VM REXX EXECs and XEDIT macros” on page 514
- “Sample files” on page 527
- “Linux code” on page 528

z/VM REXX EXECs and XEDIT macros

This section lists all of the z/VM code included in the associated tar file:

- The CPFORMAT EXEC
- The SSICMD EXEC
- PROFILE EXEC for Linux virtual machines
- The RHEL64 EXEC
- The RHEL64 EXEC

The CPFORMAT EXEC

Following is the code for the EXEC that formats multiple DASD using **CPFMTXA**. It is described in section 5.8, “Add page and perm volumes” on page 82.

```
/************************************************************/
/*                                                          */
/*  This program is provided on an "AS IS" basis, without   */
/*  warranties or conditions of any kind, either express or */
/*  implied including, without limitation, any warranties   */
/*  or conditions of title, non-infringement,               */
/*  merchantability or fitness for a particular purpose.    */
/*  Neither recipient nor any contributors shall have any   */
/*  liability for any direct, indirect, incidental,         */
/*  special, exemplary, or consequential damages (including */
```

```
/*  without limitation lost profits), however caused and on  */
/*  any theory of liability, whether in contract, strict     */
/*  liability, or tort (including negligence or otherwise)   */
/*  arising in any way out of the use or distribution of     */
/*  the program or the exercise of any rights granted        */
/*  hereunder, even if advised of the possibility of such    */
/*  damages.                                                 */
/*                                                           */
/*************************************************************/
/*                                                           */
/* Purpose:                                                  */
/*   CP format one, a range or multiple ranges of DASD.      */
/*   and label these DASDs.                                  */
/*                                                           */
/* Inputs:                                                   */
/*   dasds - address(es) of the DASD to format.              */
/*   type  - type of formatting to be done: PERM, PAGE, SPOL */
/*           or TEMP.                                        */
/*                                                           */
/* Output:                                                   */
/*   Virtual DASD that is CP formatted and labeled.          */
/*                                                           */
/* Return codes:                                             */
/*   0 - success                                             */
/*   1 - help was asked for or given                         */
/*   2 - user did not respond Y to confirm formatting        */
/*   3 - DASD (minidisk) range is not valid                  */
/*   4 - at least one DASD (minidisk) is reserved to MAINT   */
/*                                                           */
/* References:                                               */
/*  The Cloud Computing Cookbook for z/VM 6.2, RHEL 6.2 and  */
/*    SLES 11 SP2                                            */
/*  URL: http://www.vm.ibm.com/devpages/mikemac/SG248147.pdf */
/*                                                           */
/*************************************************************/
  Address COMMAND
  firstchar = 'J'
  Arg dasds 'AS ' type .
  If dasds = '' | dasds = '?' Then Call help
  labelPrefix = firstchar || getLabelPrefix(type)
  numDasd = parseDasd(dasds)
  answer = areYouSure(type)
  If answer = 'Y' Then Do
     /* the user is sure */
     formatted = ''
     retVal = doFormat(labelPrefix numDasd type)
     Call doReport retVal
     End
  Else retVal = 2
  Exit retVal

/*+------------------------------------------------------------------+*/
help:
  Procedure Expose firstchar
/*+------------------------------------------------------------------+*/
```

```
  Parse Source . . fn .
  Say
  Say 'Synopsis:'
  Say
  Say '  Format and label DASD as page, perm, spool or temp disk space'
  Say '  The label written to each DASD is' firstchar || '<t><xxxx> where:'
  Say '    <t> is type - P (page), M (perm), S (spool) or T (Temp disk)'
  Say '    <xxxx> is the 4 digit address'
  Say
  Say 'Syntax is:'
  Say "                <---------------<               "
  Say "   >>--CPFORMAT--.-vdev--------.--AS---.-PERM-.---------><"
  Say "                 '-vdev1-vdev2-'       '-PAGE-'"
  Say "                                       '-SPOL-'"
  Say "                                       '-TEMP-'"
  Say
  Exit 1

/*+------------------------------------------------------------------+*/
areYouSure:
  Procedure
/*| Warn the user of possible data loss and ask if it is okay to     |*/
/*|  format the DASD.                                                |*/
/*|  parm 1: format type for the virtual DASD                        |*/
/*|  retVal: first character of response. continue if 'Y'.          |*/
/*+------------------------------------------------------------------+*/
  Arg type
  Say
  Say 'WARNING - this will destroy data!'
  Say 'Are you sure you want to format the DASD as' type 'space (y/n)?'
  Pull answer .
  Return  'LEFT'(answer,1) /* from areYouSure */

/*+------------------------------------------------------------------+*/
getLabelPrefix:
  Procedure expose firstchar
/*| Return the second character of the virtual DASD label            |*/
/*|  parm 1: format type for the virtual DASD                        |*/
/*+------------------------------------------------------------------+*/
  Arg type .
  firstchar. = 0
  firstchar.PERM = 'M'
  firstchar.PAGE = 'P'
  firstchar.SPOL = 'S'
  firstchar.TEMP = 'T'
  If firstchar.type = 0 Then Do
  /* Incorrect formatting type specified. Provide help and quit. */
     Say 'Error: "AS" must be present, type must be PERM, PAGE, SPOL or TEMP'
     Call help
     End
  Return firstchar.type

/*+------------------------------------------------------------------+*/
parseDASD:
  Procedure Expose dasdList.
```

```
/*| parse all dasd into an array verifying all are attached          |*/
/*|  parm 1: dasds - the list of dasd passed in                       |*/
/*|  retVal: number of DASD in dasdList                               |*/
/*+------------------------------------------------------------------+*/
  Arg dasds
  numDasd = 0
  dropheader = ''
  Say
  Say 'Format the following DASD:'
  Do While dasds <> ''
     Parse Upper Var dasds dasd dasds
     dashPos = 'POS'('-',dasd)
     If dashPos = 0 Then Do
     /* There is a singleton DASD specified. */
     /* start and end of range are the same. */
        startrange = dasd
        endrange = dasd
        End
     /* process the range of DASD */
     Else Parse Var dasd startrange  '-' endrange
     Do i = 'X2D'(startrange) To 'X2D'(endrange)
        numDasd = numDasd + 1
        dasdList.numDasd = 'D2X'(i)
        'PIPE CP QUERY MDISK' dasdList.numDasd 'LOCATION',
        dropheader,
        '|CONS'
        If rc <> 0 Then Do
           Say 'Return code from QUERY MDISK =' rc
           /* If RC=40, then HCPxxx40E has been issued and msg below */
           If rc = 40 Then Say 'DASD' dasdList.numDasd 'is not attached.'
           Exit 3
           End
        Call checkReserved(dasdList.numDasd)
        dropheader = '|DROP 1'
        End
     End
  Return numDasd /* from parseDasd */

/*+------------------------------------------------------------------+*/
doFormat:
  Procedure Expose dasdList. formatted
/*| Format all DASD specified using CPFMTXA                          |*/
/*|  parm 1: labelPrefix - the two character label prefix            |*/
/*|  parm 2: numDasd - number of DASD in the array dasdList          |*/
/*|  parm 3: type - the type of DASD format                          |*/
/*|  retVal: 0 = success                                              |*/
/*+------------------------------------------------------------------+*/
  Arg labelPrefix numDasd type
  /* Save the current settings for MORE */
  Parse Value 'DIAG'('08','CP QUERY TERM') With ' MORE' morevalues ','
  'CP TERM MORE 1 1' /* Make MORE brief */

  /* Save system identifier and SSI name */
  'PIPE CP QUERY USERID | SPEC W3 | VAR systemID'
  'PIPE CP QUERY SSI | LOCATE /SSI Name/ | SPEC W3 | VAR SSIname'
```

```
  If (SSIname = "SSINAME") Then /* variable not set */
    inSSI = 'no'
  Else
    inSSI = 'yes'

  /* Iterate through all DASD in list */
  Do i = 1 to numDasd
     label = labelPrefix  || 'RIGHT'(dasdList.i,4,'0')
     retVal = formatOne(dasdList.i type label)
     If retVal <> 0 Then Do
        Say 'Error from CPFMTXA on DASD' label 'rc =' retVal
        Leave /* error - abort this format */
        End

     /* add owner info for CP owned devices */
     If (type != 'PERM') Then /* CP owned => owner info is needed */
       If (inSSI = 'yes') Then /* add owner info */
         call addOwnerInfo(dasdList.i label SSIname systemID)
       Else
         call addOwnerInfo(dasdList.i label "NOSSI" systemID)
     formatted = formatted label
     End /* Do i = */
  'CP TERM MORE' morevalues
  Return retVal /* from doFormat */

/*+------------------------------------------------------------------+*/
checkReserved:
  Procedure
/*| Try copying an already formatted DASD Then relabelling it        |*/
/*|  parm 1: dasd - the virtual address of the DASD                  |*/
/*+------------------------------------------------------------------+*/
  Arg dasd
  /* Create a list of reserved virtual DASD addresses. */
  /* Ensure that a system minidisk is not formatted.   */
  resvd = '122 123 124 190 191 193 19D 19E 2CC 401 402 990 CF1 CF3 CFD'
  If 'POS'(resvd,dasd) <> 0 Then Do
     /* MAINT minidisk - ABORT! */
     Say 'Minidisk' dasd 'is a reserved MAINT minidisk'
     Say 'This must be formatted manually using a different vaddr.'
     Exit 4
     End /* If dasd is reserved */
  Return /* from checkReserved */

/*+------------------------------------------------------------------+*/
doReport:
  Procedure Expose dasds formatted
/*| Report on the newly labelled DASD                                |*/
/*|  parm 1: formatSuccess - 0=all is well, non-0= a format failed   |*/
/*|  retVal: 0 = success                                             |*/
/*+------------------------------------------------------------------+*/
  Arg formatSuccess
  If formatSuccess <> 0 Then
    Say 'Error was encountered! retVal from CPFMTXA =' formatSuccess
  If formatted = '' Then
    Say 'No DASD were successfully formatted'
```

```
  Else
    Say 'DASD successfully formatted:' formatted
  'CP DETACH' dasds
  'CP ATTACH' dasds '*'
  Say
  Say 'DASD status after:'
  'CP QUERY MDISK' dasds 'LOCATION'
  Return 0 /* from doReport */

/*+------------------------------------------------------------------+*/
formatOne:
  Procedure
/*| Format a DASD via DDR                                             |*/
/*|  parm 1: disk - the vaddr to be formatted                         |*/
/*|  parm 2: type - PERM, PAGE, SPOL or TEMP                          |*/
/*|  parm 3: label - the six character label                          |*/
/*+------------------------------------------------------------------+*/
  Arg disk type label
  Queue 'FORMAT'
  Queue disk
  Queue '0 END'
  Queue label
  Queue 'YES'
  Queue type '0 END'
  Queue 'END'
  'EXEC CPFMTXA'
  retVal = rc
  Return retVal /* from formatOne */

/*+------------------------------------------------------------------+*/
AddOwnerInfo:
  Procedure
/*| Tag PAGE, SPOL and TDSK volumes with SSI                         |*/
/*|  parm 1: disk - the vaddr to be formatted                         |*/
/*|  parm 2: type - PERM, PAGE, SPOL or TEMP                          |*/
/*|  parm 3: label - the six character label                          |*/
/*+------------------------------------------------------------------+*/
  Arg disk label SSIname systemID
  Queue 'OWNER'
  Queue disk
  Queue label
  Queue SSIname
  Queue systemID
  'EXEC CPFMTXA'
  retVal = rc
  Return retVal /* from addOwnerInfo */
```

The SSICMD EXEC

Following is the code for the EXEC that issues CP commands on all joined members of a subsystem interface (SSI) cluster. It is recommended to reside on the MAINT 191 disk.

```
/***********************************************************/
/*                                                         */
/*  This program is provided on an "AS IS" basis, without  */
```

```
/*  warranties or conditions of any kind, either express or  */
/*  implied including, without limitation, any warranties    */
/*  or conditions of title, non-infringement,                */
/*  merchantability or fitness for a particular purpose.     */
/*  Neither recipient nor any contributors shall have any    */
/*  liability for any direct, indirect, incidental,          */
/*  special, exemplary, or consequential damages (including  */
/*  without limitation lost profits), however caused and on  */
/*  any theory of liability, whether in contract, strict     */
/*  liability, or tort (including negligence or otherwise)   */
/*  arising in any way out of the use or distribution of     */
/*  the program or the exercise of any rights granted        */
/*  hereunder, even if advised of the possibility of such    */
/*  damages.                                                 */
/*                                                           */
/*************************************************************/
/*                                                           */
/* Purpose:                                                  */
/*   Issue a command on all members of a cluster using the   */
/*   response from QUERY SSI to find the member names.       */
/*                                                           */
/* Inputs:                                                   */
/*   cmd - the CP command to issue on each member.           */
/*                                                           */
/* Output:                                                   */
/*   The results from issuing the AT command.                */
/*                                                           */
/* References:                                               */
/*  The Cloud Computing Cookbook for z/VM 6.2, RHEL 6.2 and  */
/*    SLES 11 SP2                                            */
/*  URL: http://www.vm.ibm.com/devpages/mikemac/SG248147.pdf */
/*                                                           */
/*************************************************************/
  Address COMMAND
/* The command is passed by the caller */
  Arg cmd
  /* Provide help if requested or if no command is specified */
  If cmd = '' | cmd = '?' Then Call Help
  /* Determine the members of the SSI cluster */
  'PIPE CP QUERY SSI',
  '| STEM MSG.',           /* Save the response if error */
  '| XLATE',               /* Make all output upper case */
  '| FRTARGET ALL /SLOT/', /* Just look after 'SLOT' */
  '| LOCATE /JOINED/',     /* JOINED members can do a command */
  '| SPEC W2',             /* Get the member names */
  '| STEM SSI.'            /* Save the member names */
  /* If nonzero return code, show error message and exit */
  If rc <> 0 | ssi.0 = 0 Then Do
     Say 'Error: QUERY SSI return code =' rc
     Say msg.1
     End
  Else Do
  /* Send the command to each member of the SSI cluster */
     Do i = 1 To ssi.0
        Say ssi.i||":"
```

```
      'CP AT' ssi.i 'CMD' cmd
      Say
      End
    End
  Exit

help:
  /* Provide syntax information to the user */
  Say 'SSICMD cmd'
  Say
  Say 'cmd is a command to be issued on each of the members'
  Say '  in the SSI cluster using the AT command.'
  Exit
```

PROFILE EXEC for Linux virtual machines

This section lists the code for the PROFILE EXEC that is shared among Linux virtual machines from the LNXMAINT 192 disk.

```
/* PROFILE EXEC for Linux virtual servers */
'CP SET RUN ON'
'CP SET PF11 RETRIEVE FORWARD'
'CP SET PF12 RETRIEVE'
'ACC 592 C'
'SWAPGEN 300 524288' /* create a 256M VDISK disk swap space */
'SWAPGEN 301 1048576' /* create a 512M VDISK disk swap space */
'PIPE CP QUERY' userid() '| var user'
parse value user with id . dsc .
if (dsc = 'DSC') then /* user is disconnected */
  'CP IPL 100'
else  /* user is interactive -> prompt */
 do
  say 'Do you want to IPL Linux from minidisk 100? y/n'
  parse upper pull answer .
  if (answer = 'Y') then 'CP IPL 100'
 end
```

The RHEL64 EXEC

This section lists the code for the RHEL64 EXEC that starts an RHEL 6.4 installation. It is recommended to reside on the LNXMAINT 192 disk.

```
/*************************************************************/
/* Punch a RHEL 6.4 install system to reader and IPL it      */
/* Input files: RHEL64 KERNEL, <ID> PARM-RH6, RHEL64 INITRD  */
/*************************************************************/
  Address 'COMMAND'
  'CP SPOOL PUN *'
  'CP CLOSE RDR'
  'CP PURGE RDR ALL'
  'PUNCH RHEL64 KERNEL * (NOHEADER'
  'PUNCH' 'USERID'() 'PARM-RH6 * (NOHEADER'
  'PUNCH RHEL64 INITRD * (NOHEADER'
  'CP CHANGE RDR ALL KEEP'
  'CP IPL 00C CLEAR'
```

```
  Exit
```

The SLES11S3 EXEC

This section lists the code for the sles11s3 EXEC that starts a SLES 11 SP3 installation. It is recommended to reside on the LNXMAINT 192 disk.

```
/* Punch a SLES 11 SP3 install system to reader and IPL it   */
/*************************************************************/
  Address 'COMMAND'
  'CP SPOOL PUN *'
  'CP CLOSE RDR'
  'CP PURGE RDR ALL'
  'PUNCH SLES11S3 KERNEL * (NOHEADER'
  'PUNCH' 'USERID'() 'PARM-S11 * (NOHEADER'
  'PUNCH SLES11S3 INITRD * (NOHEADER'
  'CP CHANGE RDR ALL KEEP'
  'CP IPL 00C CLEAR'
  Exit
```

The SWAPGEN EXEC

Following is the code for the EXEC that creates Linux swap spaces from z/VM VDISKs.

```
/*******************************************************************
* Program: SWAPGEN EXEC
*
* Original Author: Dave Jones (djones@sinenomine.net)
*
* Description/Purpose:
*
* Generate VDISK swap for Linux on System z guest virtual
* machines
*
* Syntax:
*
* Issue: SWAPGEN ? for syntax etc.
*
* Version History:
...
*/

   address command

   arg vdev blks . '(' options ')'

   debug = 0                                        /* Default to quiet */
   fba = 0                                         /* No FBA option yet */
   reuse = 0                                     /* No reuse option yet */
   do while options <> ''                          /* Parse the options */
      parse var options option options                 /* Get an option */
      select
         when option = 'DIAG' then fba = 0         /* Use DIAG driver */
         when option = 'FBA' then fba = 1           /* Use FBA driver */
         when option = 'REUSE' then reuse = 1           /* Reuse DASD */
```

```
      when option = 'DEBUG' then debug = 1    /* Wants debug chat */
      when option = 'VERSION' then signal Version /* version query*/
      otherwise
      say 'Invalid option "'option'"'             /* Else unknown */
   end
end

minblks = 40 - 8 * fba  /* Minimum number of blocks that can work */

if reuse = 1 then do
   parse value diagrc(8, 'Q V 'vdev) ,     /* Get blocks from ... */
     with rc . 17 msg                    /* ... actual device size */
   if rc <> 0 then signal BadDev
   parse var msg . . . . . newblks .
   if blks = '' then blks = newblks   /* Default to detected size */
   if blks <> newblks then signal WrongBlks    /* Mismatch, error */
end

if vdev = '?' then signal Help            /* Wants Help, give it */
if vdev = '' then signal NoVdev                /* Missing, error */
if blks = '' then signal NoBlks                /* Missing, error */
if datatype(blks, 'W') = 0 | blks < minblks then /* Bad/too small */
signal BadBlks                                       /* So error */
if datatype(vdev, 'X') = 0 | length(vdev) > 4 then     /* Invalid */
signal BadVdev                                       /* So error */

if fba then do    /* If FBA driver, make sure we have the package */
   'NUCEXT RXDASD'                            /* Already got it?? */
   if rc <> 0 then 'NUCXLOAD RXDASD'        /* No, try to load it */
   if debug then say 'SWAPGEN: Loading RXDASD got rc=' rc
   if rc <> 0 then signal NoRXDASD       /* That failed, so error */
end

if reuse = 0 then do
   call diag 8, 'DETACH' vdev       /* DETACH any existing device */
   parse value diagrc(8, 'DEFINE VFB-512 AS' vdev 'BLK' blks) ,
     with rc . 17 msg '15'x                    /* Define the V-DISK */
   if debug then say 'SWAPGEN: DEFINE VDEV got rc=' rc
   if rc <> 0 then signal BadDefine      /* That failed, so error */
end
call csl 'DMSGETFM rc reacode  fm'        /* find a free filemode */
if debug then say 'SWAPGEN: Got filemode' fm 'from DMSGETFM'
if rc <> 0 then signal NoFreeModes /* Weren't any, strange, error */

if fba then do
   pages = trunc((blks * 512)/4096) - 1              /* FBA case */
   writeit = 'stem swap.'                          /* Pipe stage */
   if debug then say 'SWAPGEN: Computed' pages 'for FBA disk'
end
else do                  /* Not FBA, we must FORMAT and RESERVE it */
   'MAKEBUF'          /* Guard stack contents if something's there */
   buf = rc    /* Remember buffer number so we drop the right one */
   if debug then say 'SWAPGEN: Acquired buffer' buf ,
     'before non-FBA format.'
   writeit = 'specs number 1 1-* next' ,
```

```
      '| mdskupdate LINUX SWAP' fm 'F 512'         /* Pipes stages */
    queue '1'                          /* Yes to the format? question */
    queue 'LXSWAP'                                /* Disk volume name */
    queue '1'                          /* Yes to the reserve question */
    'PIPE (name SWPFORMAT)' ,          /* FORMAT and RESERVE the disk */
      '| cms FORMAT' vdev fm '(BLKSIZE 512 NOERASE' ,    /* FORMAT */
      '| var rs1' ,                       /* Remember how that went */
      '| hole' ,                          /* And otherwise pitch it */
      '| cms RESERVE LINUX SWAP' fm ,                      /* Do it */
      '| var rs2' ,                       /* Remember how that went */
      '| hole' ,                          /* And otherwise pitch it */
      '| state LINUX SWAP' fm  , /* Look at the reserved swap file */
      '| var reserveok' ,                  /* Keep that information */
      '| specs w6 1'          ,  /* Word 6 is the number of blocks */
      '| specs w1 1' ,                     /* Calculate it modulo 8 */
      'a: word 1 .' ,                      /* Get the token we want */
      '   set #0:=a%8-2' ,        /* Calculate it modulo 8 minus 2 */
      '   print #0 20' ,                                /* Write it */
      '| specs w2 1' ,      /* Just get the number of usable pages */
      '| var pages'                           /* And remember that */
    if debug then say 'SWAPGEN: Formatted' pages 'pages on disk' ,
      fm 'in PIPE'
    'DROPBUF' buf          /* Not nice to leave trash lying around */
    if debug then say 'SWAPGEN: Dropped buffer' buf
  end

  if debug then say 'SWAPGEN: About to write non-FBA swap signature'

/* Must use separate Pipe to write since mdskupdate commits to 0 */
  'PIPE (name SWPWrite)' ,
    '| var pages' ,                           /* Get number of pages */
    '| specs pad 00 w1 d2c 1.4 right' ,                /* Format it */
    '| append strliteral x'c2x(copies('00'x, 4086-1033+1) || ,
    '53574150535041434532'x) ,           /* "SWAPSPACE2" in ASCII */
    '| join' ,                       /* Build that into a nice chunk */
    '| preface strliteral x'c2x(copies('00'x, 1027)'01'x) ,   /* 0s */
    '| join' ,                       /* Build that into a nice chunk */
    '| deblock 512',                          /* Break into records */
    '|' writeit   /* And write to disk or variable, per driver type */
  if debug then say 'SWAPGEN: Wrote non-FBA swap signature with rc=' rc
  if rc <> 0 then signal BadWrite

/* If FBA, we have the values, need to use RXDASD to write them */
  if fba then do i = 1 to swap.0     /* If FBA, we didn't write yet */
    if debug then say 'SWAPGEN: About to write FBA signature' i
    rc = DASD('WRITED', vdev, i-1, swap.i)           /* Write one */
    if debug then say 'SWAPGEN: Wrote FBA signature' i 'with rc='rc
    if rc <> 0 then signal BadWrite           /* Failed, so error */
  end

  if fba then type ='FBA'
  else type = 'DIAG'
  say type 'swap disk defined at virtual address' vdev , /* Success! */
    '('pages-1' 4K pages of swap space)'
  call Quit 0
```

```
Quit:
   arg rc
   if rc <> 0 then say 'No Swap disk was created.'
   exit rc

NoVdev:              /* User didn't give us a virtual device address */
   say 'A virtual device address must be specified!'
   signal Help

NoBlks:                    /* User didn't give us a number of blocks */
   say 'Number of blocks must be specified!'
   signal Help

NoFreeModes:                      /* No free disk modes can be found */
   say 'No free disk modes are available!'
   say 'Please release a minidisk and try again.'
   call Quit 1      /* They invoked it correctly, so don't show help */

BadDev:                             /* REUSE tried to use bad device */
   say 'The device at 'vdev' cannot be used:'
   say msg
   call Quit 24

BadBlks:                 /* User gave us an invalid number of blocks */
   say 'Invalid number of blocks "'blks'" specified; must be'
   say 'at least 'minblks' 512-byte blocks.'
   call Quit 24

WrongBlks:               /* Supplied number of blocks does not match */
   say 'REUSE requested with' blks ,
     'and existing disk block count is' newblks'.'
   call Quit 24

BadVdev:           /* User gave us an invalid virtual device address */
   say 'Invalid virtual device address "'vdev'" specified;'
   say 'must be a 1- to 4-digit hexadecimal value.'
   call Quit 24

NoRXDASD:                  /* We don't have the required FBA utility */
   say 'Unable to NUXCLOAD RXDASD MODULE; this is available from:'
   say ' http://www.vm.ibm.com/download/packages'
   call Quit rc

BadDefine:                             /* Error DEFINE-ing the VDISK */
   say 'Error' rc 'from CP DEFINE VFB-512 AS' vdev 'BLK' blks':'
   say msg                                /* Display error from CP */
   call Quit rc

BadFBA:                         /* Error writing FBA block on disk */
   say 'Error' rc 'from RXDASD'
   call Quit rc

BadWrite:                        /* Error on FORMAT or RESERVE steps */
   select                          /* Figure out where it went wrong */
```

```
      when symbol('RESERVEOK') <> 'VAR' then do
         say 'Error' rc 'from CMS RESERVE LINUX SWAP' fm':'
         say rs2
      end
      when symbol('RS2') <> 'VAR' then do
         say 'Error' rc 'from CMS FORMAT' vaddr fm '(BLKSIZE 512:'
         say rs1
      end
      otherwise
      say 'Error' rc 'calculating swap size, contact support'
   end
   call Quit rc

Help:
   parse source . . fn .

   say 'Syntax is:'
   say ''
   say fn 'vdev #blocks <( <options> <)> >'
   say '   or'
   say fn 'vdev ( REUSE <options> <)>'
   say ''
   say 'where:'
   say ''
   say 'vdev     -- is a virtual device address'
   say '#blocks  -- is a decimal number of 512-byte blocks;'
   say '            minimum 24 (FBA) or 32 (DIAG)'
   say ''
   say 'Options are:'
   say 'DIAG     -- (Default) Use DIAG I/O (requires Linux DIAG driver)'
   say 'FBA      -- use FBA driver instead of DIAG; requires RXDASD'
   say '            package, downloadable from the IBM VM download'
   say '            page at: http://www.vm.ibm.com/download/packages'
   say 'REUSE    -- use existing device at vdev.  WARNING: This will'
   say '            destroy any data on device vdev.  The #blocks'
   say '            parameter may be omitted; the whole device will'
   say '            be used in that case.'
   say 'VERSION  -- display current version number string and date'
   say '            of last module update.'
   say 'DEBUG    -- display progress messages and debugging'
   say '            information about the program logic. '
   say ''
   say fn 'will DETACH any existing virtual device at that address,'
   say 'DEFINE a new VDISK, format it, and write the Linux swap'
   say 'signature on the disk so Linux will recognize it.'
   say ''
   say 'If using FBA mode, SWAPGEN prepares the whole device:'
   say '    /dev/dasdb or /dev/dasd/0151/device'
   say 'so the whole device must then specified in the Linux fstab.'
   say ''
   say 'If using DIAG mode, because the V-DISK is CMS FORMATted,'
   say 'SWAPGEN prepares the partition:'
   say '    /dev/dasdb1 or /dev/dasd/0151/part1'
   say 'so the partition must be specified in the fstab on Linux.'
   call Quit 1
```

```
Version:
   parse source . . fn .

/* These variables should be updated with each release */
   version = 'SNA120601'                /* Release string: SNAyymmvv */
   last_update = '2012-06-20 (yyyy-mm-dd)'       /* Last update date */

   say 'SWAPGEN: Version' version', last updated:' last_update'.'
   call Quit 4
```

Sample files

This section lists sample files described in the book.

The SAMPLE CONF-RH6 file

This section lists the sample RHEL 6 configuration file:

```
DASD=100-103,300-301
HOSTNAME=hostName.DNSname.com
NETTYPE=qeth
IPADDR=n.n.n.n
SUBCHANNELS=0.0.0700,0.0.0701,0.0.0702
NETMASK=255.255.255.0
SEARCHDNS=DNSname.com
GATEWAY=n.n.n.n
DNS=n.n.n.n
MTU=1500
PORTNAME=DONTCARE
PORTNO=0
LAYER2=1
```

The SAMPLE PARM-RH6 file

This section lists the sample RHEL 6 configuration file:

```
root=/dev/ram0 ro ip=off ramdisk_size=40000
CMSDASD=191 CMSCONFFILE=userid.CONF-RH6
vnc vncpassword=12345678
```

The SAMPLE PARM-S11 file

This section lists the sample SLES 11 SP3 configuration file:

```
ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dumb
HostIP=n.n.n.n Hostname=yourhost.example..com
Gateway=n.n.n.n Netmask=255.255.255.0
Broadcast=n.n.n.n Layer2=1
ReadChannel=0.0.0700  WriteChannel=0.0.0701  DataChannel=0.0.0702
Nameserver=n.n.n.n
portname=whatever
portno=0
```

```
Install=nfs://n.n.n.n/var/nfs/sles11sp3/SLES-11-SP2-DVD-s390x-GM-DVD1.iso
UseVNC=1  VNCPassword=12345678
InstNetDev=osa OsaInterface=qdio OsaMedium=eth Manual=0
```

Linux code

This section contains listings of the following Linux scripts:

- The RHEL clone script
- The SLES clone.sh script
- The SLES boot.clone script

The RHEL clone script

This section lists the code for the **/usr/sbin/clone** script that clones from an RHEL golden Linux image to a target virtual machine. It is contained in the RPM `clone-1.0-11.s390x.rpm`.

```
#!/bin/sh
#
# clone.sh is a script that clones Linux images.  It makes use of vmcp to
# relay messages to the z/VM system and configuration files to modify
# the new image once it has been cloned.
#
# The script reads in /etc/sysconfig/clone for user setting customizations.
#
# For details on how this script works see the book:
#   "z/VM and Linux on IBM System z: The Virtualization Cookbook for RHEL4"
#   on the Web at: http://www.redbooks.ibm.com/abstracts/sg247272.html
#
# -----------------------------------------------------------------------------
# THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY
# WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY
# OR FITNESS FOR A PARTICULAR PURPOSE.
# NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY
# DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
# (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY
# OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
# NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR
# DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED
# HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES
# -----------------------------------------------------------------------------

# These MUST be lower case!
MASTER_LINK=fffe
CLONE_LINK=ffff

#+----------------------------------------------------------------------------+
function help
# give help
#+----------------------------------------------------------------------------+
 {
  echo "Usage: clone [-v] sourceID targetID [rootMinidisk [minidisk1
minidisk2..]]"
```

```
  echo "    Switches"
  echo "       -v Verbose output"
  echo "    Required"
  echo "       sourceID the z/VM user id you want to clone from"
  echo "       targetID  the z/VM user id you want to clone to"
  echo "    Optional"
  echo "       rootMinidisk the minidisk address that contains the root
filesystem"
  echo "       minidisk1..n additional minidisks that should be copied"
  exit
 }

#+--------------------------------------------------------------------------+
function cp_cmd
# echo a CP command and invoke it via cp_cmd
#   Arg1-n: the z/VM command to issue
#   Return: the z/VM command's return code
#+--------------------------------------------------------------------------+
 {
  [ -n "$VERBOSE" ] && echo "Invoking CP command: $@"
  out=$(vmcp $@ 2>&1)
  rc=$?

  # Pull the z/VM error code from the output
  if [ $rc -ne 0 ] ; then
   rc=$(echo $out | grep Error | sed s/.*#//g)
   [ -z "$rc" ] && rc=1
  fi
  return $rc
 }

#+--------------------------------------------------------------------------+
function copy_key
# If the host has a id_dsa.pub file then append that to the clone's
# authorized_keys file.
#+--------------------------------------------------------------------------+
 {
  if [ -e /root/.ssh/id_dsa.pub ] ; then
    [ ! -d /mnt/clone/root/.ssh/ ] && mkdir -p /mnt/clone/root/.ssh/
    echo "# LNXINST" >> /mnt/clone/root/.ssh/authorized_keys
    cat /root/.ssh/id_dsa.pub >> /mnt/clone/root/.ssh/authorized_keys
    chmod 600 /mnt/clone/root/.ssh/authorized_keys
  fi
 }

#+--------------------------------------------------------------------------+
function abort
# Exit the script and clean up
#+--------------------------------------------------------------------------+
 {
  umount_cloned_image

  set_offline $CLONE_LINK
  set_offline $MASTER_LINK
```

```
  unlink_one $CLONE_LINK
  unlink_one $MASTER_LINK

  exit $1
 }

#+--------------------------------------------------------------------------+
function get_target_info
# Get the TCP/IP and DNS info for the Linux ID to clone to.  This function
# will check both the shared.conf file and the specific target id's conf
# file.  If values are still missing then the user will be prompted to
# supply them.
#+--------------------------------------------------------------------------+
 {
  unset HOSTNAME
  [ -f /etc/clone/shared.conf ] && . /etc/clone/shared.conf
  [ -f /etc/clone/${target_linux_id}.conf ] && .
/etc/clone/${target_linux_id}.conf

  shift # drop the MasterGuestID
  shift # drop the CloneGuestID

  # If there are still command line arguments then the user must have specified
DASD
  # on the command line.  Unset whatever we have in DASD (from the config
files) and
  # set DASD equal to the rest of the arguments.
  [ $# -gt 0 ] && DASD="$@" && unset DASD_ROOT

  # Loop through all of the values that we require and double check that they
have
  # values.  If they don't then we will prompt the user to fill them in.
  for v in HOSTNAME IPADDR DNS GATEWAY NETMASK MTU SUBCHANNELS SEARCHDNS
NETTYPE DASD
  do
   if [ -z "$(eval echo \$$v)" ]; then
    [ "$PROMPT" != "y" ] && echo "Error: missing required value for $v" && exit
1
    [ -z "$first" ] && echo "Please enter $target_linux_id's value for: " &&
first=1
    echo -n "$v: "
    read in
    eval $(echo $v=\"$in\")
    export $v
    echo "$v=$in" >> /etc/clone/${target_linux_id}.conf
   fi
  done

  # Expand DASD ranges if they have been defined
  if [ -n "$DASD" ] ; then
   split=$(echo $DASD | tr ',' ' ')
   DASD=""
   for s in $split
   do
    out=$(echo $s | grep \-)
```

```
    rc=$?
    [ $rc -eq 0 ] && DASD=${DASD}$(seq -s" " $(echo $s | tr '-' ' ' | tr '\n' '
'))
    [ $rc -ne 0 ] && DASD=${DASD}$(echo -n "$s ")
   done
   [ -n "$DASD_ROOT" ] && DASD=$(echo $DASD | sed "s/$DASD_ROOT//")
   DASD="$DASD_ROOT $DASD"
   # Assuming that if no DASD_ROOT is specified then the first DASD device will
be
   # take as root
   if [ -z "$DASD_ROOT" ] ; then
     DASD_ROOT=$(echo $DASD | awk -F" " '{print $1}')
   fi
   export DASD
  fi

  # Grab just the hostname with out any DNS suffixes from the FQDN
  target_host=$(echo $target_fqhost | awk -F. '{print $1}')
 }


#+--------------------------------------------------------------------------+
function dd_copy
# Use the dd command to copy one disk to another
#   Arg 1: Source minidisk - assumed to be online
#   Arg 2: Target minidisk - must be brought online and dasdfmt'd
#+--------------------------------------------------------------------------+
 {
  ret_val=0

  source_mdisk=$1
  target_mdisk=$2

  # Bring the source and target devices online
  set_online $source_mdisk
  set_online $target_mdisk

  target_dev_node=`cat /proc/dasd/devices | grep "$target_mdisk(ECKD)" | awk '{
print $7 }'`
  source_dev_node=`cat /proc/dasd/devices | grep "$source_mdisk(ECKD)" | awk '{
print $7 }'`

  wait_for_device /dev/$target_dev_node
  ret_val=$?

  if [ $ret_val -eq 0 ] ; then
    [ -n "$VERBOSE" ] && echo "Invoking Linux command: dasdfmt -p -b 4096 -y -F
-f /dev/$target_dev_node"
    [ -n "$VERBOSE" ] && progress="-p"
    dasdfmt $progress -b 4096 -y -F -f /dev/$target_dev_node
    [ $? -ne 0 ] && echo "Error: dasdfmt failed" && ret_val=1
  fi

  if [ $ret_val -eq 0 ] ; then
    wait_for_device /dev/$source_dev_node
```

```
    ret_val=$?
  fi

  if [ $ret_val -eq 0 ] ; then
    nblks=`cat /proc/dasd/devices | grep $target_dev_node | awk '{ print $13
}'`
    [ -n "$VERBOSE" ] && \
    echo "Invoking Linux command: dd bs=4096 count=$nblks
if=/dev/$source_dev_node of=/dev/$target_dev_node"
    dd bs=4096 count=$nblks if=/dev/$source_dev_node of=/dev/$target_dev_node
>/dev/null
    [ $? -ne 0 ] && echo "Error: dd failed" && ret_val=1
  fi

  # Put the source and target devices offline
  set_offline $target_mdisk
  set_offline $source_mdisk

  return $ret_val
 }

#+--------------------------------------------------------------------------+
function link_one
# This will link one minidisk from another user id as the target minidisk
# address on the current z/VM user id with a link mode indicated by the
# 4th argument.
#
#   Arg1: Source z/VM ID
#   Arg2: Source minidisk virtual address
#   Arg3: Target minidisk virtual address
#   Arg4: Link mode (rr/w)
#+--------------------------------------------------------------------------+
 {
  source_id=$1
  source_mdisk=$2
  target_mdisk=$3
  link_mode=$4

  cp_cmd QUERY VIRTUAL $target_mdisk
  if [ $? != 40 ]; then
    cp_cmd DETACH $target_mdisk
  fi

  cp_cmd LINK $source_id $source_mdisk $target_mdisk $link_mode $LINK_PASSWD
  if [ $? != 0 ]; then
    echo "cp_cmd link $source_id $source_mdisk $target_mdisk $link_mode failed
- exiting"
    abort 1
  fi
  }

#+--------------------------------------------------------------------------+
function unlink_one
# This will unlink a minidisk from the current z/VM user id.
#   Arg1: The target minidisk to unlink
```

```
#+--------------------------------------------------------------------------+
 {
  cp_cmd DETACH $1
  return $?
 }

#+--------------------------------------------------------------------------+
function copy_one
# Try to use z/VM FLASHCOPY to copy one disk to another. If that fails,
#    call dd_copy() to fall back to the Linux DD command
#   Arg 1: Source minidisk
#   Arg 2: Target minidisk
#+--------------------------------------------------------------------------+
 {
  source_mdisk=$1
  target_mdisk=$2

  if [ "$CLONE_METHOD" == "AUTO" -o "$CLONE_METHOD" == "auto" ] ; then
    cp_cmd FLASHCOPY $source_mdisk 0 END $target_mdisk 0 END
    rc=$?
    if [ $rc -ne 0 ]; then # FLASHCOPY failed
      [ -n "$VERBOSE" ] && echo "FLASHCOPY $source_mdisk $target_mdisk failed
with $rc - using Linux dd"
    else
     return 0
    fi
  fi

  dd_copy $source_mdisk $target_mdisk
  [ $? -ne 0 ] && return 1
 }

#+--------------------------------------------------------------------------+
function copy_disks
# Call copy_one to copy each disk passed in as an argument.
#   Arg1-n: The minidisk address to copy
#+--------------------------------------------------------------------------+
 {
  [ -n "$VERBOSE" ] && echo "Copying minidisks..."
  while [ $# -gt 0 ]; do
    link_one $source_linux_id $1 $MASTER_LINK RR
    link_one $target_linux_id $1 $CLONE_LINK W
    copy_one $MASTER_LINK $CLONE_LINK
    [ $? -eq 0 ] && echo "$1 disk copied ..."
    unlink_one $MASTER_LINK
    unlink_one $CLONE_LINK
    shift
  done
 }

#+--------------------------------------------------------------------------+
function link_disks
# Call link_one to link each disk passed in as an argument.
#   Arg1-n: The minidisk address to link
#+--------------------------------------------------------------------------+
```

```
 {
  [ -n "$VERBOSE" ] && echo "Linking minidisks for LVM..."
  while [ $# -gt 0 ]; do
    link_one $target_linux_id $1 400$# W
    set_online 400$#
    [ $? -eq 0 ] && echo "$1 disk linked ..."
    shift
  done
 }

#+--------------------------------------------------------------------------+
function unlink_disks
# Call unlink_one to unlink each disk passed in as an argument.
#   Arg1-n: The minidisk address to unlink
#+--------------------------------------------------------------------------+
 {
  [ -n "$VERBOSE" ] && echo "Unlinking minidisks ..."
  while [ $# -gt 0 ]; do
    set_offline 400$#
    unlink_one 400$#
    [ $? -eq 0 ] && echo "$1 disk unlinked ..."
    shift
  done
 }

#+--------------------------------------------------------------------------+
function ask_are_you_sure
# Ask "Are you sure?" - if not, then exit
#+--------------------------------------------------------------------------+
 {
  echo ""
  echo "This will copy disks from $source_linux_id to $target_linux_id"
  echo "Host name will be:  $HOSTNAME"
  echo "IP address will be: $IPADDR"
  echo -n "Do you want to continue? (y/n): "
  read ans
  if [ $ans != "y" ]; then
    abort 1
  fi
 }

#+--------------------------------------------------------------------------+
function check_logged_off
# Verify the user ID exists and is logged off
#   Arg1: The user id to query if it is logged on or not
#+--------------------------------------------------------------------------+
 {
  cp_cmd QUERY $1
  case $? in
    0)  # user ID is logged on or disconnected
      echo "$1 user ID must be logged off"
      exit 2
      ;;
    3)  # user ID does not exist
      echo "$1 user ID does not exist"
```

```
      exit 3
      ;;
   45) # user ID is logged off - this is correct
      ;;
   *) # unexpected
      echo "$1 user ID must exist and be logged off"
      exit 4
  esac
 }

#+--------------------------------------------------------------------------+
function modify_cloned_image
# Modify the networking information in appropriate files under /etc
# Regenerate SSH keys in golden image's /etc/ssh/ directory and change root pw
#+--------------------------------------------------------------------------+
 {
  source_ipaddr=$(grep IPADDR
$CLONE_MNT_PT/etc/sysconfig/network-scripts/ifcfg-eth0 \
        | awk -F= '{print $2}')
  source_hostname=$(grep HOSTNAME $CLONE_MNT_PT/etc/sysconfig/network \
        | awk -F= '{print $2}')
  source_host=$(echo $source_hostname| awk -F. '{print $1}')

  [ ! -d $CLONE_MNT_PT/etc ] && echo "Error: no $CLONE_MNT_PT/etc found" &&
abort 1

  [ -n "$VERBOSE" ] && echo "Modifying networking info under $CLONE_MNT_PT..."
  sed -i \
    -e "s/$source_ipaddr/$IPADDR/g" \
    -e "s/$source_hostname/$HOSTNAME/g" \
    -e "s/$source_host/$target_host/g" \
  $CLONE_MNT_PT/etc/hosts

  sed -i \
    -e "s/HOSTNAME=.*/HOSTNAME=$HOSTNAME/g"\
    -e "s/GATEWAY=.*/GATEWAY=$GATEWAY/g"\
  $CLONE_MNT_PT/etc/sysconfig/network

  sed -i \
    -e "s/IPADDR=.*/IPADDR=$IPADDR/g"\
    -e "s/MTU=.*/MTU=$MTU/g"\
    -e "s/NETMASK=.*/NETMASK=$NETMASK/g"\
    -e "s/SUBCHANNELS=.*/SUBCHANNELS=$SUBCHANNELS/g"\
    -e "s/NETTYPE=.*/NETTYPE=$NETTYPE/g"\
  $CLONE_MNT_PT/etc/sysconfig/network-scripts/ifcfg-eth0

  # Modify MACADDR/HWADDR if specified (optional)
  [ -n "$MACADDR" ] && sed -i -e "s/MACADDR=.*/MACADDR=$MACADDR/g" \
  $CLONE_MNT_PT/etc/sysconfig/network-scripts/ifcfg-eth0

  [ -n "$HWADDR" ] && sed -i -e "s/HWADDR=.*/HWADDR=$HWADDR/g" \
  $CLONE_MNT_PT/etc/sysconfig/network-scripts/ifcfg-eth0

  # Regenerate the SSH keys on the new clone's root filesystem
```

```
  [ -n "$VERBOSE" ] && echo "Regenerating SSH keys in $CLONE_MNT_PT/etc/ssh/
..."
  rm -f $CLONE_MNT_PT/etc/ssh/ssh_host*
  ssh-keygen -t rsa -N "" -q -f $CLONE_MNT_PT/etc/ssh/ssh_host_rsa_key
  ssh-keygen -t dsa -N "" -q -f $CLONE_MNT_PT/etc/ssh/ssh_host_dsa_key
  ssh-keygen -t rsa1 -N "" -q -f $CLONE_MNT_PT/etc/ssh/ssh_host_key

  copy_key
 }


#+--------------------------------------------------------------------------+
function set_online
# This will set online the target minidisk.
#   Arg1 - Minidisk virtual address to set online
#+--------------------------------------------------------------------------+
 {
  local target_mdisk=$(echo $1 | tr 'A-Z' 'a-z')
  chccwdev -e 0.0.$target_mdisk >/dev/null
  rc=$?
  if [ $rc != 0 ]; then
    echo "Error: chccwdev -e 0.0.$target_mdisk failed with $rc - exiting"
    abort 1
  fi

  local target_dev_node=`cat /proc/dasd/devices | grep "$target_mdisk(ECKD)" |
awk '{ print $7 }'`
  if [ "$target_dev_node" = "" ]; then
    echo "Error: can't find $target_mdisk(ECKD) in /proc/dasd/devices -
exiting"
    set_offline $target_mdisk
    abort 1
  fi
 }

#+--------------------------------------------------------------------------+
function set_offline
# This will set offline the target minidisk.
#   Arg1 - Minidisk virtual address to set offline
#+--------------------------------------------------------------------------+
 {
   target_mdisk=$(echo $1 | tr 'A-Z' 'a-z')
   chccwdev -d 0.0.$target_mdisk > /dev/null 2>&1
   rc=$?
   #if [ $rc -ne 0 ]; then
   # echo "Error: chccwdev -d 0.0.$1 failed with $rc - ignoring"
   #fi

   return $rc
 }

#+--------------------------------------------------------------------------+
function mount_cloned_image
# This will mount the cloned root filesystem.  It will pair a minidisk
# address to a device file and then mount the first partition.
```

```
#   Arg1: The minidisk address to mount
#+--------------------------------------------------------------------------+
 {
  target_mdisk=$1

  target_dev_node=`cat /proc/dasd/devices | grep "$target_mdisk(ECKD)" | awk '{
print $7 }'`

  wait_for_device /dev/${target_dev_node}1
  [ $? -ne 0 ] && echo "Error: timed out waiting for /dev/${target_dev_node}1"
&& abort 1

  /bin/mount /dev/${target_dev_node}1 $CLONE_MNT_PT
  [ $? -ne 0 ] && echo "Error: unable to mount cloned image" && abort 1

  /bin/mount | grep /dev/${target_dev_node}1 >/dev/null 2>&1
  [ $? -ne 0 ] && echo "Error: unable to mount cloned image" && abort 1

 }

#+--------------------------------------------------------------------------+
function mount_cloned_image_lvm
# This will mount the cloned root filesystem.  It will pair a minidisk
# address to a device file and then mount the first partition.
#   Arg1: The minidisk address to mount
#+--------------------------------------------------------------------------+
 {
  target_mdisk=$1

  /bin/mount /dev/$VG_NAME/$LV_ROOT $CLONE_MNT_PT
  [ $? -ne 0 ] && echo "Error: unable to mount cloned image" && abort 1

  /bin/mount | grep $LV_ROOT >/dev/null 2>&1
  [ $? -ne 0 ] && echo "Error: unable to mount cloned image" && abort 1

 }

#+--------------------------------------------------------------------------+
function umount_cloned_image
#   Unmount the cloned root filesystem
#+--------------------------------------------------------------------------+
 {
  /bin/umount $CLONE_MNT_PT >/dev/null 2>&1

  return $?
 }

#+--------------------------------------------------------------------------+
function check_for_conf
# Check that the configuration file exists for the ID that we are cloning to.
#+--------------------------------------------------------------------------+
 {
  if [ ! -f /etc/clone/${target_linux_id}.conf -a "$PROMPT" != "y" ]; then
   echo "Error: /etc/clone/${target_linux_id}.conf not found. Exiting"
   exit
```

```
  fi
 }

#+--------------------------------------------------------------------------+
function check_for_vmcp
# Check that the vmcp module is loaded and the vmcp binary is installed.
#+--------------------------------------------------------------------------+
 {
   # Check that vmcp exists and is executable
   [ ! -x /sbin/vmcp ] && echo "Error: can't find /sbin/vmcp" && exit

   # Load the vmcp kernel module if not already loaded
   if ! /sbin/lsmod | grep vmcp > /dev/null 2>&1 ; then
     if ! /sbin/modprobe vmcp > /dev/null 2>&1 ; then
       echo "Error: unable to load module vmcp, check kernel version"
       exit
     fi
   fi

  wait_for_device /dev/vmcp
  [ $? -ne 0 ] && echo "Error: timed out waiting for /dev/vmcp" && exit
 }

#+--------------------------------------------------------------------------+
function wait_for_device
# Sleep until a certain file exists
#   Arg1: The path of the file to sleep on.
#+--------------------------------------------------------------------------+
 {
  device=$1

  sleep 2
  for t in $(seq 1 20)
  do
    [ -e $device ] && return 0
    sleep 1
  done
  return 1
}

#+--------------------------------------------------------------------------+
function autolog
# Issue an XAUTOLOG command to bring up the new cloned image.
#+--------------------------------------------------------------------------+
 {
  cp_cmd XAUTOLOG $target_linux_id
  rc=$?
  if [ $? != 0 ]; then
    echo "xautolog $target_linux_id failed with $rc"
    return 0
  fi
  echo "Booting $target_linux_id"
 }

#+--------------------------------------------------------------------------+
```

```
# main()

# Only root can run this script
[ $(id -u) != "0" ] && echo "Error: you must be root" && exit

# Check if the user has defined any clone.sh configurations
[ -f /etc/sysconfig/clone ] && . /etc/sysconfig/clone

# Set defaults for clone.sh configurations
[ -z "$PROMPT" ] && PROMPT="y"
[ -z "$CLONE_MNT_PT" ] && CLONE_MNT_PT="/mnt/clone"

# If the clone mount point does not exist then we'll create it for you
[ ! -d $CLONE_MNT_PT ] && mkdir -p $CLONE_MNT_PT

# Check if -v was specified on the command line
if [ "$1" = "-v" ] ; then
  VERBOSE=1
  shift
fi

# If no command line options were provided show the help message
[ $# -eq 0 ] && help

# If one comand line option was provided show the help message
if [ $# -lt 2 ]; then
  echo "Error: incorrect number of arguments"
  help
fi

# Check that vmcp exists and the module is loaded
check_for_vmcp

# Allow UPPER or lower case source, target, blacklist entries.
# Convert all to lower case for consistency.
source_linux_id=$(echo $1 | tr "[:upper:]" "[:lower:]")
target_linux_id=$(echo $2 | tr "[:upper:]" "[:lower:]")

# Check the blacklist, which prevents using the master image as a target.
if [ -f /etc/clone/blacklist.conf ]; then
  . /etc/clone/blacklist.conf
  BlackList=$(echo ${BLACKLIST} | tr "[:upper:]" "[:lower:]")
  for Target in ${BlackList}
  do
    if [ "${Target}" == "${target_linux_id}" ]; then
      echo "${target_linux_id} is blacklisted!  Exiting!"
      exit
    fi
  done
fi

# Check that the master and clone z/VM IDs are logged off.
check_logged_off $source_linux_id
check_logged_off $target_linux_id
```

```
# Check that the clone's configuration file exists
check_for_conf

# Collect information from the clone's configuration file
get_target_info $@
[ "$PROMPT" = "y" ] && ask_are_you_sure

echo "Cloning $source_linux_id to $target_linux_id ..."
[ -z "$DASD" ] && echo "Error: no DASD defined in
/etc/clone/${target_linux_id}.conf" && exit
copy_disks $DASD

# Update the newly cloned image locally, so link, set online then mount the
# clone's root filesystem.  Then call modify_cloned_image to update
# configuration files with the proper settings.  Finally unmount,
# set offline and unlink the disk.
echo "Updating cloned image ..."
if [ -n "$VG_NAME" ]; then
  link_disks $DASD
  # FIXME wait for disks
  sleep 2
  /sbin/vgscan
  # FIXME wait for vgscan
  sleep 2
  /sbin/vgchange -a y $VG_NAME
  mount_cloned_image_lvm $CLONE_LINK
else
  link_one $target_linux_id $DASD_ROOT $CLONE_LINK W
  set_online $CLONE_LINK
  mount_cloned_image $CLONE_LINK
fi
modify_cloned_image
umount_cloned_image
if [ -n "$VG_NAME" ]; then
  /sbin/vgchange -a n $VG_NAME
  unlink_disks $DASD
else
  set_offline $CLONE_LINK
  unlink_one $CLONE_LINK
fi

# Autolog the clone unless AUTOLOG has been set to "n"
[ "$AUTOLOG" = "y" ] && autolog

echo "Successfully cloned $source_linux_id to $target_linux_id"
```

The SLES clone.sh script

This section lists the code for the **/usr/local/sbin/clone.sh** script that clones from a SLES golden Linux image to a target virtual machine.

```
#!/bin/sh
#
# clone.sh <LinuxUserID> - clone a Linux server running under z/VM
#
```

```
# For details on how this script works see the book:
#  "z/VM and Linux on IBM System z: The Cloud Computing Cookbook
#    for z/VM 6.3 RHEL 6.2 and SLES 11 SP3"
#  on the Web at: http://www.vm.ibm.com/devpages/mikemac/CKB-VM62.pdf
#
# -----------------------------------------------------------------------------
# THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY
# WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY
# OR FITNESS FOR A PARTICULAR PURPOSE.
# NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY
# DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
# (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY
# OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
# NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR
# DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED
# HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES
# -----------------------------------------------------------------------------

#+----------------------------------------------------------------------------+
function help()
# give help
#+----------------------------------------------------------------------------+
 {
  echo "Usage: clone [options] from <sourceID> to <targetID>"
  echo ""
  echo "  Clone Linux from sourceID 100 and 101 minidisks to targetID"
  echo "  options:"
  echo "    -v or --verbose: verbose"
  echo ""
  echo "Example: clone.sh from s11gold to linux01"
  exit 1
 }

#+----------------------------------------------------------------------------+
function processArguments()
# Parse command line arguments
# Args: The arguments passed in to the script
#+----------------------------------------------------------------------------+
 {
  verbose="off"
  sourceID="none"
  targetID="none"
  while (( "$#" )); do
    case $1 in
      -v|--verbose)
         verbose="on"
         ;;
      from)
        shift
        sourceID=`echo $1 | tr '[a-z]' '[A-Z]'` # fold source ID to upper case
        ;;
      to)
        shift
        targetID=`echo $1 | tr '[a-z]' '[A-Z]'` # fold target ID to upper case
```

```
        ;;
    esac
    shift
  done
  if [ $sourceID = "none" ]; then # source user ID was not passed
    echo "Error: Source Linux user ID not supplied"
    help
  fi
  if [ $targetID = "none" ]; then # target user ID was not passed
    echo "Error: Target Linux user ID not supplied"
    help
  fi
 }

#+--------------------------------------------------------------------------+
function CPcmd()
# echo a CP command and invoke it via the vmcp module/command
#   Arg1-n: the command to issue
#   Return: the command's return code
#+--------------------------------------------------------------------------+
 {
  echo "Invoking CP command: $@"
# parse output to get return code: awk -F# splits line at '#' with rc at end
  output=`vmcp $@ 2>&1`
  echo "$output"
  retVal=0
  retVal=`echo $output | grep "Error: non-zero CP response" | awk -F# '{print
$2}'`
  return $retVal
 }

#+--------------------------------------------------------------------------+
function checkID()
# Verify user ID exists and is logged off
#    Arg 1: The user ID to check
#+--------------------------------------------------------------------------+
 {
  userID=$1
  echo "Checking that $userID exists and is not logged on ..."
  CPcmd QUERY $userID
  rc=$?
  case $rc in
    0)  # user ID is logged on or disconnected
      echo "$userID user ID must be logged off"
      exit 2
      ;;
    3)  # user ID does not exist
      echo "$userID user ID does not exist"
      exit 3
      ;;
   45) # user ID is logged off - this is correct
      ;;
   *) # unexpected
      echo "Return code of $rc unexpected from QUERY $userID"
      echo "User ID must exist and be logged off"
```

```
      exit 4
  esac
 }

#+--------------------------------------------------------------------------+
function prepareIPaddr()
# Set the variable "newIPaddr" by adding a backslash before any "."s
#    Arg 1: The IP address to be modified
#+--------------------------------------------------------------------------+
 {
  newIPaddr=`echo $1 | sed -e 's:\.:\\\.:g'`
 }

#+--------------------------------------------------------------------------+
function prepareVaddr()
# Prepare an address by folding to lower case and prepending leading zeros
# to make it 4 digits
#    Arg 1: The vaddr to be modified
# Return:
#    The new value is written to the global variable newVaddr
#+--------------------------------------------------------------------------+
 {
  newVaddr=`echo $1 | tr '[A-Z]' '[a-z]'`  # fold to lower case
  let leadingZeros=4-${#1}                  # determine number of zeros to add
  let i=0
  while [ $i -lt $leadingZeros ]; do
    newVaddr="0$newVaddr"
    i=$[$i+1]
  done
 }

#+--------------------------------------------------------------------------+
function copyDisk()
# Use FLASHCOPY to copy a disk, if it fails, fall back to dasdfmt then dd
#   Arg 1: Source vaddr
#   Arg 2: Target vaddr
#+--------------------------------------------------------------------------+
 {
  source=$1
  target=$2
  echo ""
  echo "FLASHCOPYing $source to $target ..."
  CPcmd FLASHCOPY $source 0 end to $target 0 end
  if [ $? != 0 ]; then
    echo "FLASHCOPY failed, falling back to dasdfmt and dd ..."
    chccwdev -e $source
    if [ $? != 0 ]; then exit 7; fi
    chccwdev -e $target
    if [ $? != 0 ]; then exit 8; fi
    sleep 1
    srcDev=/dev/$(egrep ^0.0.$source /proc/dasd/devices | awk '{ print $7 }')
    if [ "$?" != 0 ]; then exit 5; fi
    tgtDev=/dev/$(egrep ^0.0.$target /proc/dasd/devices | awk '{ print $7 }')
    if [ "$?" != 0 ]; then exit 6; fi
    echo "dasdfmt-ing $tgtDev ..."
```

```
    dasdfmt -y -b 4096 -f $tgtDev
    if [ "$?" != 0 ]; then exit 9; fi
    echo "dd-ing $srcDev to $tgtDev ..."
    dd bs=1M if=$srcDev of=$tgtDev oflag=sync
    if [ "$?" != 0 ]; then exit 10; fi
    sync
    echo "disabling and re-enabling $target ..."
    chccwdev -d $target
    if [ $? != 0 ]; then exit 11; fi
    chccwdev -e $target
    if [ $? != 0 ]; then exit 12; fi
    sync
  fi
 }

#+--------------------------------------------------------------------------+
function askAreYouSure()
# Ask "Are you sure?" - if not, then exit
#+--------------------------------------------------------------------------+
 {
  echo ""
  echo "WARNING!!: Minidisks 100 and 101 will be copied to $targetID"
  echo "Network data is retrieved from $targetID PARM-S11 on 191 disk"
  echo "during the first boot of $targetID"
  echo -n "Are you sure you want to overwrite these disks (y/n): "
  read ans
  if [ $ans != "y" ]; then
    echo "Aborting clone per user input"
    exit 16
  fi
 }

#+--------------------------------------------------------------------------+
function copySystem()
# For each of two minidisks 100 and 101:
#   -) Link disk
#   -) Enable disk
#   -) Copy disk
#+--------------------------------------------------------------------------+
 {
  echo "Linking source and target 100 disks ..."
  CPcmd detach 1100
  CPcmd link $sourceID 100 1100 rr
  if [ $? != 0 ]; then exit 17; fi
  CPcmd detach 2100
  CPcmd link $targetID 100 2100 mr
  if [ $? != 0 ]; then exit 18; fi
  echo "Copying 100 disks ..."
  copyDisk 1100 2100
  echo "Take 1100 Offline...."
  chccwdev -d 1100
  CPcmd det 1100
  CPcmd det 2100

  echo " "
```

```
  echo "---------------------------------------"
  echo "Linking source and target 101 disks ..."
  CPcmd detach 1101
  CPcmd link $sourceID 101 1101 rr
  if [ $? != 0 ]; then exit 19; fi
  CPcmd detach 2101
  CPcmd link $targetID 101 2101 mr
  if [ $? != 0 ]; then exit 20; fi
  echo "Copying 101 disks ..."
  copyDisk 1101 2101
  echo "Taking 1101 Offline..."
  chccwdev -d 1101
  CPcmd det 1101
  echo "Taking 2101 Offline..."
  chccwdev -d 2101
  CPcmd det 2101
 }

# main()
processArguments $@                       # process arguments passed by user
if [ $verbose = "on" ]; then set -vx; fi  # turn on debug
checkID $sourceID                         # user ID must exist and be logged
off
checkID $targetID                         # user ID must exist and be logged
off
# getNetworkInfo                            # get info from parm files
askAreYouSure                             # confirm disks will be overwritten
copySystem                                # copy source disks to target
# modifyClone                               # modify newly copied system
echo "sleeping 10 seconds"
sleep 10
CPcmd XAUTOLOG $targetID                  # bring new clone to life
if [ $verbose = "on" ]; then set +vx; fi  # turn off debug
echo "Successfully cloned $sourceID to $targetID"
exit 0
```

The SLES boot.clone script

This section lists the code for the **/etc/init.d/boot.clone** script that runs at “first boot” of a newly cloned SLES system.

```
#!/bin/bash
#
# /etc/init.d/boot.clone
#
### BEGIN INIT INFO
# Provides:          boot.clone
# Required-Start:    boot.localfs boot.rootfsck
# Required-Stop:     boot.localfs
# Default-Start:     B
# Default-Stop:
# Short-Description: Change configuration during boot
# Description:       Change the current configuration of the system
#  during first bootup. This script works as follows:
#  1. Run vmcp q userid
```

```
#  2. Search for a cms file called userid() PARM-S11
#  3. Get new values for network config from there
#  4. Update the network configuration accordingly
#  This previously used to be the cloning.sh script on linuxadmin.
### END INIT INFO

. /etc/rc.status

rc_reset

#+--------------------------------------------------------------------------+
function CPcmd()
# echo a CP command and invoke it via the vmcp module/command
#   Arg1-n: the command to issue
#   Return: the command's return code
#+--------------------------------------------------------------------------+
 {
# echo "Invoking CP command: $@"
# parse output to get return code: awk -F# splits line at '#' with rc at end
  output=`vmcp $@ 2>&1`
  echo "$output"
  retVal=0
  retVal=`echo $output | grep "Error: non-zero CP response" | awk -F# '{print
$2}'`
  return $retVal
 }

#+--------------------------------------------------------------------------+
function prepareVaddr()
# Prepare an address by folding to lower case and prepending leading zeros
# to make it 4 digits
#    Arg 1: The vaddr to be modified
# Return:
#    The new value is written to the global variable newVaddr
#+--------------------------------------------------------------------------+
 {
  newVaddr=`echo $1 | tr '[A-Z]' '[a-z]'`  # fold to lower case
  let leadingZeros=4-${#1}                  # determine number of zeros to add
  let i=0
  while [ $i -lt $leadingZeros ]; do
    newVaddr="0$newVaddr"
    i=$[$i+1]
  done
 }

#+--------------------------------------------------------------------------+
function getUserid()
# Read current userid with vmcp q userid
#+--------------------------------------------------------------------------+
 {
 modprobe vmcp
 UserID=$(CPcmd q userid | awk '{print $1}')
 echo $UserID
 }
```

```
#+--------------------------------------------------------------------------+
function getNetworkInfo()
# Bring 191 minidisk online to check for my parameter files
#+--------------------------------------------------------------------------+
 {
  # recycle 191 to pick up latest changes
  chccwdev -d 191
  chccwdev -e 191
  rc=$?
  if [ $rc != 0 ]; then # unable to enable 191 disk
    echo "unable to enable 191, rc from chccwdev = $rc"
    exit 13
  fi
  udevadm settle
  CMSdisk=`lsdasd | grep 0191 | awk '{ print $3 }'`
  cmsfslst -d /dev/$CMSdisk | grep -i $1 | grep PARM-S11
  rc=$?
  if [ $rc != 0 ]; then
    echo "Error: $1 PARM-S11 not found on 191 minidisk. Exiting"
    exit 14
  fi

# get information about target
  { while read parameter; do
    #echo "parameter: ${parameter%=*}"
    case "${parameter%=*}" in
      Hostname)
        targetHostname=${parameter#*=}
        ;;
      HostIP)
        targetIP=${parameter#*=}
        ;;
      Nameserver)
        targetDNS=${parameter#*=}
        ;;
      Gateway)
        targetGW=${parameter#*=}
        ;;
      Netmask)
        targetMask=${parameter#*=}
        ;;
      Broadcast)
        targetBroadcast=${parameter#*=}
        ;;
      ReadChannel)
        prepareVaddr ${parameter#*=}
        targetReaddev=$newVaddr
        ;;
      WriteChannel)
        prepareVaddr ${parameter#*=}
        targetWritedev=$newVaddr
        ;;
      DataChannel)
        prepareVaddr ${parameter#*=}
        targetDatadev=$newVaddr
```

```
        ;;
      *)
        # don't know about any other parameters
        ;;
    esac
  done < <(cmsfscat -a -d /dev/$CMSdisk $1.PARM-S11 | tr '[:space:]' '\n')
  }
 }

#+--------------------------------------------------------------------------+
function createNetworkConfig()
# - remove existing network configuration if it exists
# - create new network configuration from information in CMS parmfile
# - update HOSTNAME, hosts, and resolv.conf
#+--------------------------------------------------------------------------+
 {
 # delete old configuration
 rm -f /etc/sysconfig/network/ifcfg-eth0
 # setup new configuration
 if [ -n "${targetHostname}" ]; then
   echo "Setting hostname to ${targetHostname}"
   echo ${targetHostname} > /etc/HOSTNAME
 fi
 if [ -n "${targetDNS}" ]; then
   echo "Setting dns resolver to ${targetDNS}"
   sed -i '/nameserver/d' /etc/resolv.conf
   echo "nameserver ${targetDNS}" >> /etc/resolv.conf
 fi
 # echo target stuff
 # will add configuration of different devices when time permits.
 if [ -n "${targetIP}" ]; then
   echo "Setting IP address to ${targetIP}"
   echo "STARTMODE='onboot'" >> /etc/sysconfig/network/ifcfg-eth0
   echo "BOOTPROTO='static'" >> /etc/sysconfig/network/ifcfg-eth0
   echo "IPADDR='${targetIP}'" >> /etc/sysconfig/network/ifcfg-eth0
 fi
 if [ -n "${targetMask}" ]; then
   echo "Setting netmask to ${targetMask}"
   echo "NETMASK='${targetMask}'" >> /etc/sysconfig/network/ifcfg-eth0
 fi
 if [ -n "${targetBroadcast}" ]; then
   echo "Setting broadcast to ${targetBroadcast}"
   echo "BROADCAST='${targetBroadcast}'" >> /etc/sysconfig/network/ifcfg-eth0
 fi
 if [ -n "${targetGW}" ]; then
   echo "Setting default gateway to ${targetGW}"
   sed -i '/default/d' /etc/sysconfig/network/routes
   echo "default ${targetGW} - -" >> /etc/sysconfig/network/routes
 fi
 }
#+--------------------------------------------------------------------------+
function cleanupSSH()
# - remove all existing ssh keys
#+--------------------------------------------------------------------------+
 {
```

```
# Delete SSH keys - sshd will recreate them at first boot
  echo "Removing SSH keys"
  rm /etc/ssh/ssh_host*
 }


case "$1" in
    start)
   # update system configuration
        userid=$(getUserid)
        getNetworkInfo $userid
        createNetworkConfig
        cleanupSSH
        chkconfig boot.clone off
   rc_reset
   ;;
    stop|restart)
        # this should never happen
        # nothing to do
   ;;
    status)
   # probably never will be run.
        # nothing to do
   ;;
    *)
   echo "Usage: $0 {start}."
   exit 1
   ;;
esac

rc_exit
```

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

The following IBM Redbooks publications provide additional information about the topic in this document. Note that some publications referenced in this list might be available in softcopy only:

- *Linux on IBM eServer zSeries and S/390: Performance Toolkit for VM*, SG24-6059
- *z/VM and Linux on IBM System z*, SG24-7492
- *Linux on IBM eServer zSeries and S/390: Application Development*, SG24-6807
- *IBM Lotus Domino 6.5 for Linux on zSeries Implementation*, SG24-7021
- *Printing with Linux on zSeries Using CUPS and Samba*, REDP-3864

You can search for, view, download, or order these documents and other Redbooks, Redpapers, Web Docs, draft and additional materials, at the following website:

ibm.com/redbooks

Other publications

These publications are also relevant as further information sources:

- *z/VM Performance Toolkit Guide*, SC24-6156-00
- *z/VM Performance Toolkit Reference*, SC24-6157-00

Online resources

These websites are also relevant as further information sources:

- The Linux for zSeries and S/390 portal:

  http://linuxvm.org
- The IBMVM list server:

  http://listserv.uark.edu/archives/ibmvm.html
- The Linux-390 list server:

  http://www2.marist.edu/htbin/wlvindex?linux-390
- SUSE Linux Enterprise Server 9 evaluation:

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

- z/VM publications:

  http://www.vm.ibm.com/pubs
- z/VM performance tips:

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

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

The Virtualization Cookbook for IBM z/VM 6.3, RHEL 6.4, and SLES 11 SP3

(1.0” spine)
0.875”<->1.498”
460 <-> 788 pages

The Virtualization Cookbook for IBM z/VM 6.3, RHEL 6.4, and SLES 11 SP3

A guide to install and customize z/VM, Linux, and open source appliances

Updated for the most current releases

Detailed, step-by-step instructions

This IBM Redbooks publication describes how to *roll your own* Linux virtual servers on IBM System z hardware under IBM z/VM. It adopts a cookbook format that provides a concise, repeatable set of procedures for installing and configuring z/VM 6.3 into a single system image (SSI), then installing and customizing Linux.

You need at least two IBM System z logical partitions (LPARs) with associated resources, z/VM 6.3 media, and either the Red Hat Enterprise Linux (RHEL) version 6.4 or the SUSE Linux Enterprise Server (SLES) version 11 SP3 distribution (or both).

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, and to get some virtual servers up and running in a short amount of time (days, not weeks or months).

SG24-8147-00 ISBN 0738438936