Lotus software

IBM

# IBM Lotus Domino 6.5 for Linux on zSeries Implementation

- Planning the environment - operating system, disk, network
- Installing and administering Linux and Domino
- Capacity planning and performance

Mike Ebbers, Wanda Brewster
Barbara Filippi, Mike Wojton
Faheem Ashraf, Andreas Gallus
Chris Henry, Danny Lawrence
Cathleen Fox Leger, Brian Macfaden
Mike MacIsaac, Elsie Ramos
Don Robbins, Barton Robinson

Redbooks

ibm.com/redbooks

International Technical Support Organization

# IBM Lotus Domino 6.5 for Linux on zSeries Implementation

December 2004



SG24-7021-01

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



**Second Edition (December 2004)**

This edition applies to Release 6.5 of IBM Lotus Domino for Linux on zSeries (product number AH0IJNA), SuSE Linux Enterprise Server version 8 (SLES-8) with service pack 2, and Version 4.3 of IBM z/VM.

# Contents



# Notices

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

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

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

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

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

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

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

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

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

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

# Trademarks

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

@server®
e-business Software™
eServer™
ibm.com®
iNotes™
iSeries™
z/Architecture™
z/OS®
z/VM®
zSeries®
AIX®
CICS®
DirMaint™
Domino Designer®
Domino®
DB2®
Enterprise Storage Server®
ECKD™
ESCON®
FICON®
HiperSockets™
Hummingbird®
IBM®
Lotus Notes®
Lotus®
Multiprise®
MQSeries®
MVS™
Notes®
OfficeVision/MVS™
OfficeVision/VM™
OfficeVision®
OS/2®
OS/390®
OS/400®
Redbooks™
Redbooks(logo) ™
RAMAC®
RMF™
S/360™
S/390®
Sametime®
Tivoli®
VM/ESA®
VSE/ESA™
WebSphere®

The following terms are trademarks of other companies:

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

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

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

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

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

# Preface

This IBM® Redbook will help the reader plan for, install, and configure the new IBM Lotus® Domino® 6.5 product on an IBM @server® zSeries® system running Linux. We discuss why you should consider running Domino for Linux on zSeries, and list the advantages of running Linux in a guest under z/VM®. We then describe the structure of running Domino for Linux on zSeries for those who are not familiar with all of those products, and review what is new in Domino 6.5, since this is the first release that is supported on Linux on zSeries.

We provide detailed technical information about planning, allocating, and managing disk space, network considerations, installing Linux and Domino, and administering Domino. We also discuss systems management, capacity planning and performance tuning, connectivity to DB2®, migration from previous Domino releases or Domino servers on other platforms, and troubleshooting.

This redbook is targeted to zSeries systems programmers, Domino administrators, consultants, and service technicians.

This updated version includes information from customer and IBM experiences.

## The team that wrote this redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization, Poughkeepsie Center.

**Mike Ebbers** is the Project Leader for this redbook (which was developed with generous help from the authoring team that follows). Mike is a Consulting IT Specialist at the International Technical Support Organization (ITSO), Poughkeepsie Center. He has been with IBM for 29 years, spending ten years in technical marketing, eleven years in technical education, and eight years in the ITSO. He leads redbook projects on Lotus products and on IBM products such as workflow and imaging.

**Wanda Brewster** is a Consulting IT Specialist with the Domino for zSeries Team in the Washington Systems Center. She worked on both the first and second edition of this book. She has worked at IBM for 25 years and has more than 10 years of experience with Domino and Notes. Her areas of expertise include Domino installation and administration, and migration from and coexistence with legacy office systems such as OfficeVision/VM™, OfficeVision/MVS™, and DISOSS.

**Barbara Filippi** is a Consulting IT Specialist with the Domino for zSeries Team in the Washington Systems Center. She updated the second edition of this book. She has worked at IBM for 25 years and has been involved with Domino on zSeries since it initially became available on that platform. Her areas of expertise include Domino installation and administration, capacity planning, performance analysis, and migration to zSeries from other Domino platforms.

**Mike Wojton** is a Consulting IT Specialist with the Domino for zSeries Team in the Washington Systems Center. He updated the second edition of this book. He has worked at IBM for 22 years and has been involved with Domino on zSeries since its initial release on 4.5.1. His areas of expertise with Domino include installation, capacity planning, and performance analysis.

**Faheem Ashraf** is an Advisory IT specialist with Server Group zSeries from Denmark. He holds a master's degree in Science from Technical University of Denmark. He has worked at IBM for seven years and has more than five years of experience with Domino on zSeries. His current responsibilities include zSeries pre-sale support for Nordics. His areas of expertise include Domino installation and administration, as well as migration and consolidation to Domino on zSeries.

**Andreas Gallus** is an IT Specialist for IBM Global Services in Hamburg, Germany. He has over five years of experience in Domino, along with UNIX and z/OS®. He holds a Ph.D. in Physics from University of Heidelberg. His areas of expertise include Domino infrastructure and administration, as well as planning for and performing Domino server migrations (R4 to R5, R4/R5 to Domino 6), and consolidation to Domino for zSeries.

**Chris Henry** is a Senior IT Specialist from Milwaukee working in Field Support Services. He has been with IBM for 11 years, joining as part of the Lotus acquisition. Before joining Field Support services, he supported the Softswitch product line in the field, as well as Notes and Domino. He has extensive experience with most messaging solutions on the market, including Exchange, Groupwise, cc:Mail, OfficeVision®, and SMTP messaging. Prior to joining IBM, he spent 12 years working in operations and as a Senior Systems Programmer supporting the VM and MVS operating systems. He also has a background in UNIX and Linux. His areas of expertise include Domino installation, messaging, and Domino transaction logging, and he has presented extensively on Domino transaction logging and backup solutions.

**Danny Lawrence** is the President of Tiassa Technologies, Inc., an IBM Business Partner concentrating on Lotus Notes® and Domino infrastructure, deployment, and management. He has been working with Notes and Domino since 1993, specializing in infrastructure and deployment issues, with a special interest in Notes security. He has published several articles in trade magazines.

**Cathleen Fox Leger** is an Advisory Software Engineer with IBM Technical Support in North Reading, MA. She has over eight years of experience with IBM/Lotus Domino, and previous experience in enterprise IT management. She holds a bachelor's degree from the College of St. Rose, Albany, N.Y., along with diplomas from the University of Fribourg, Switzerland and the Institute for Computer Studies, Toronto, Ontario. Her current focus is providing world-class Domino for zSeries technical support to her customer and internal contacts. Cathleen has written and reviewed many technical documents in the Lotus Software Knowledge Base.

**Brian Macfaden** is an IBM @server Consultant in the United Kingdom. He has 26 years of experience in IBM. He worked as a Project Leader at the ITSO for five years, where he produced four Redbooks™ on Domino for S/390®. He has also presented on Domino for S/390 around the world. He currently provides technical pre-sales support to customers across IBM's server and storage range, advising on optimizing the IT infrastructure and cost of ownership. Brian holds a degree in Mathematics from Oxford University.

**Mike MacIsaac** performs marketing technical support on zSeries, focusing on Linux and z/VM. He has led and worked on many redbook projects.

**Elsie Ramos** is a Senior IT Specialist for IBM Global Services in Poughkeepsie, N.Y. She holds a master's degree in Computer Science from Pace University. She has worked at IBM for 20 years, and has 10 years of experience with Lotus Notes/Domino and with VM Support. She has seven years of experience in the deployment of Domino on zSeries, starting from November 1996 when the first pilot server was installed in Poughkeepsie. Her areas of expertise also includes Domino administration on various platforms including AIX®, OS/2®, Windows®, and Linux.

**Don Robbins** is a Systems Engineer for Velocity Software in the US. He has 18 years of experience in the VM, UNIX and Linux fields, and specializes in performance. He previously contributed to the redbook *Linux on IBM @server zSeries and S/390: Performance Measurement and Tuning*, SG24-6926.

**Barton Robinson** is President of Velocity Software, Inc. He began working with VM in 1975, specializing in performance starting in 1983. His previous publication experience includes the IBM publication *VM/HPO Tuning Guide*, the Velocity Software publication *VM/ESA Tuning Guide*, and the Redbooks *Linux on IBM @server zSeries and S/390: ISP/ASP Solutions*, SG24-6299, and *Linux on IBM @server zSeries and S/390: Performance Measurement and Tuning*, SG24-6926. He is the author and developer of ESAMAP and ESATCP.

Special thanks to the following people for working with the authoring team:

**Sreehari Haridevara** has been with IBM Global Services for more than six years, working on Domino on zSeries. He piloted, and currently supports, the first production Domino on Linux on zSeries server at IBM.

**Jeff Wells** has been working with Linux on zSeries for more than three years. He currently supports various production applications including Domino for Linux on zSeries.

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

Roy Costa
Rich Conway
Bob Haimowitz
Dave Bennin
International Technical Support Organization, Poughkeepsie Center

Bill Bitner
Romney White
IBM Endicott

Rob van der Heij
IBM Netherlands

Don Corbett
IBM Poughkeepsie

Laura Bostian
Thomas Bradley
Gary Chapman
Jim Dewan
Clark Goodrich
Wu W. Huang
Domino development on zSeries, IBM Poughkeepsie

## Become a published author

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

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

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

**ibm.com**/redbooks/residencies.html

## Comments welcome

Your comments are important to us!

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

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

  **ibm.com**/redbooks

- Send your comments in an Internet note to:

  redbook@us.ibm.com

- Mail your comments to:

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

1

# Introduction

In this redbook, we describe how to plan for, install, and manage the IBM Lotus Domino server on the Linux operating system on zSeries hardware.

Implementing this solution requires the involvement of people who are skilled in various areas—Domino, Linux, zSeries hardware, and optionally the z/VM operating system.

This redbook addresses all of those areas, and different readers may want to refer to different parts of the book. However, we strongly recommend using a team approach to the implementation, since good communication is *essential* to a successful result.

In this chapter, we cover the following:

- Reasons to run Domino for Linux on zSeries
- Domino's platform independence
- Structure of Domino for Linux on zSeries
- What's new in Domino 6.5

## 1.1 Contents and audience

In this chapter, we discuss why you should consider running Domino for Linux on zSeries, and list the advantages of running Linux in a guest under z/VM. We then describe the structure of running Domino for Linux on zSeries for those who are not familiar with all of those products, and review what is new in Domino 6.5, since this is the first release that is supported on Linux on zSeries.

This chapter is intended for zSeries systems programmers, Domino administrators, consultants, and service technicians, as well as managers, planners, and decision makers.

The following chapters provide more detailed technical information about planning, allocating and managing disk space, network considerations, installing Linux and Domino and administering Domino. We then discuss systems management, capacity planning and performance tuning, connectivity to DB2, and migration from previous Domino releases or Domino servers on other platforms.

### 1.1.1 Scope and assumptions

We assume that you already have z/VM installed and working, if you will be using it to support Linux and Domino. We discuss the installation of Linux, either natively, or in an LPAR, or under z/VM, and cover the advantages of those environments in more detail in various sections of this chapter.

We also assume that you have done the necessary planning work for Domino, either by having an existing Domino domain that you will add the server into, or by completing a detailed Domino planning exercise. We assume that you have already decided on names for servers, domains, administrators, and so on. This work will require an experienced Domino administrator.

Following installation of the Domino server, we describe how to administer and manage it, focusing on differences and best practices specific to Domino for Linux on zSeries, and we refer you to other Domino books for information about basic Domino administration.

Refer to “Related publications” on page 397 for help with tasks not covered in this redbook.

## 1.2 Reasons to run Domino for Linux on zSeries

Here we discuss why you might consider running Lotus Domino on Linux on the zSeries platform. This section is aimed at managers, decision makers, and planners. For more detailed information about the structure of the solution, see 1.4, “Structure of Domino on Linux on zSeries” on page 6.

### 1.2.1 Domino on Linux

The Linux operating system is becoming very popular for many applications. Some advantages of using Linux for Domino are:

- It allows you to change the cost model for delivering the Domino service. While Linux can be downloaded from the Internet for free, it is likely that you will pay for a Linux distribution, and for Linux support and application software. However, it changes the cost model because Linux runs on many platforms in the marketplace, so you can select the platform that best fits your needs in terms of quality of service (such as scalability and availability), cost, and risk.

- It provides a great deal of flexibility, since you can move applications to different hardware platforms without changing them.
- It is based on UNIX, and therefore has scalability, availability, and security more like UNIX than Windows.

### 1.2.2 Value of the zSeries platform

Having decided to run Domino on Linux, the zSeries servers have advantages over other server types:

- zSeries servers are the most reliable and scalable servers for mission-critical applications in the marketplace.
- You can add capacity to the server while the system is running; this is called Capacity Upgrade on Demand. This allows you to start with a small zSeries server and add capacity as the number of users increases.
- A single zSeries server can host many Domino servers spread across multiple images of Linux, thus enabling a single server to support tens of thousands of Notes users and applications.
- With this scalability, zSeries can provide a lower total cost of ownership. With fewer hardware servers and Domino servers, you benefit from less complexity and a lower cost of administration and management.
- zSeries LPAR technology leads the industry. Its ability to share CPUs between LPARs make it possible to reduce cost by delivering unused processing cycles in one LPAR to an LPAR that requires additional processing cycles. zSeries also makes it possible to configure more LPARs than the number of physical CPUs.
- Linux on zSeries provides a very scalable Domino solution by using the new sys_epoll system call in United Linux 1.0 to significantly improve Linux scalability.

### 1.2.3 Value of z/VM

Linux will run directly on zSeries server hardware, either natively on the whole machine, or in a logical partition (or LPAR; refer to “zSeries logical partitioning (LPAR)” on page 9 for more information).

Another alternative is to run the z/VM operating system on the server (either the whole server or in an LPAR,) and then run Linux in one or more z/VM virtual machines. The additional benefits of running on z/VM are:

- You can run dozens (even hundreds) of Linux servers on a single system under z/VM, far more than the LPAR method would allow. This enables you to rapidly consolidate many Domino servers to zSeries without needing to change your Domino infrastructure
- It is very easy to create and delete virtual Linux guest machines, if you have a need to do that on a regular basis. It is also very simple to move them from one VM system to another, if those systems are sharing DASD.
- z/VM provides facilities for system management across multiple Linux servers.

### 1.2.4 Comparison of Domino on zSeries—Linux and z/OS

With Domino 6.5 there are now two ways to run Domino on a zSeries platform: either on the z/OS operating system, or on the Linux operating system (with or without z/VM). It is not the purpose of this book to recommend one or the other, since the preferred option for you will depend on your specific needs and circumstances.

However, it may be helpful to list some of the different characteristics of these two operating systems as far as Domino is concerned. We have listed these under the three headings that we think you should consider—quality of service, cost, and risk.

## Quality of service (QoS)

By quality of service, we mean the standard of service that you are able to deliver to the users, including reliability, availability, serviceability (RAS), scalability, security, and manageability. Linux and z/OS differ in these areas:

- While both operating systems benefit from the quality of service features of the zSeries hardware, Domino on z/OS exploits more of the hardware features, such as use of the hardware cryptographic facility that improves SSL performance.
- z/OS contains many advanced functions for running mission-critical applications. Half of the operating system kernel is there for error recovery, so the z/OS operating system seldom fails. Many administration tasks can be done without taking the operating system down.
- z/OS contains many advanced features for automated system management, including advanced workload management with Intelligent Resource Director, Resource Measurement Facility for performance monitoring, and System Management Facility for resource consumption measurement.
- z/OS has been developed over many years to run efficiently on a 32-way processor. Linux is not proven to these levels of scalability, which means that for a large system you will need to run more versions of the operating system. z/VM is proven to run efficiently on many processors, so you could run a single version of z/VM with multiple versions of Linux in separate virtual machines.
- Domino on z/OS is proven to run thousands of Domino users in a single DPAR under z/OS. At the time of writing, Linux is not proven to the same levels of scalability.
- DECS for Linux on zSeries (Linux or z/OS) currently supports only connections to DB2.

## Cost

A second consideration is cost. This redbook cannot provide detailed cost comparisons, but we suggest you keep in mind the following points:

- Hardware costs depend on detailed sizing information that is not available at the time of writing. IBM offers IFLs for Linux, in addition to standard zSeries processors. Disk costs are likely to be the same in both options. Keep in mind that hardware costs are likely to be less than one-third of the total cost of running the solution over five years.
- Software costs may not differ between z/OS and Linux as much as you would expect. Domino is chargeable in both environments. While the z/OS cost is likely to be higher than Linux, options such as z/OSe on z800 servers (for small to medium installations) can reduce that cost. You will pay for a Linux distribution, and will also probably pay for Linux support. In both cases you will need some additional software, such as for backup.

  z/OS costs can limited by using NALC or Workload License Charge, or by running Domino on a separate zSeries server from other zSeries workloads to avoid having to license unnecessary products on Domino capacity. On Linux, other software costs can be minimized by using IFLs, because if you use standard processors, you may again incur additional costs for other products on z/OS on the same machine.
- People and support costs are likely to be the highest cost that you incur in running the Domino service. The additional hardware and software costs in z/OS provide you with a great deal of extra function that will reduce your support effort and therefore your personnel costs. Refer to “Quality of service (QoS)” on page 4 for a list of these capabilities, and consider how they will reduce support costs.

- Also consider the costs of training and building up experience. Using skills that you have today will avoid the costs of developing or bringing in new skills. If you do not have skills today, Linux skills are more readily available in the market than z/OS skills.

### Risk

The third factor to consider is risk. A higher risk project is likely to result in a lower quality of service, at least initially, and a higher cost to resolve unforeseen issues. Factors to consider include:

- What level of scalability and reliability do you require? For the highest levels, Domino on z/OS has been proven in the marketplace since 1997. Domino on Linux first became available on Intel® servers in 2000, and first became available on zSeries in 2003.
- What skills do you have available? z/OS is a more complex operating environment. If you have z/OS skills now, adding Domino to that environment is relatively simple, but if you do not have those skills, developing them will take time. If you have Linux skills, or UNIX skills, then implementing Domino on Linux should be straightforward.
- What is your preferred cultural approach to running Domino? If you have, or want to have, centralized control, then Domino on z/OS will minimize the number of Domino servers you need to run. If you want to continue with decentralized control, as perhaps on Windows today, then Domino on Linux will allow you to run many small servers efficiently, especially under z/VM.

In short, z/OS offers the ultimate in zSeries reliability and scalability. You can choose to run Domino on either z/OS or on Linux by considering a number of factors, especially what skills you have available.

## 1.3 Domino's platform independence

Questions are sometimes asked about differences in features and functions of Domino on various platforms; these questions might be phrased as:

- "What do I lose if move Domino to a non-Windows operating system?"
- "What do I gain if I run Domino on a UNIX-type operating system?"

The answer is: nothing; that is, Domino is full-functioned and platform-independent across all platforms. We asked a Domino developer to comment on this in more detail; he responded with an explanation in two parts: source code, and application environment.

### 1.3.1 Domino source code

The Domino source is made up of multiple layers of code. The top-most layers of code are platform-independent and are developed on whichever architecture the developer is most familiar with (UNIX, PC, or Mac), and tested on all architectures. These layers use no platform-specific code.

There is another layer which is client-side only, and this is coded on Windows with an emulation layer providing the identical functionality on the Mac.

Finally, there is the OS-specific layer. This layer consists of a platform-independent API which is coded and created for each supported architecture of Domino, and gives Domino access to features which are platform-dependent in a platform-neutral behavior.

For example, OSLoadProgram is used by Domino to load a program with no knowledge of the underlying architecture. The code for OSLoadProgram itself abstracts out the architectural

differences to the caller, even though it itself is doing platform-specific operations. When Domino is ported to an additional platform, it is *this* OS layer which must be ported and which is the majority of the development work for the port. (There are always exceptions to this, usually specific to compiler behaviors.)

Of significant note, it can actually be the QE effort which takes up the largest amount of time during a port due to the fact that the entire product must be tested on the new platform, and with millions of lines of code, this can and does take a considerable amount of time.

### 1.3.2 Domino API application porting

The complexity of porting a Domino C/C++ API application to a new platform is for the most part dependent on how the program was initially designed and created. If the program was created in such a way as to use the Domino OS layer API for all of its access to platform-specific functionality, then the development portion of the port should be minor.

It is when API programs use platform-specific code within their product instead of using the Domino OS API that the development portion of the port becomes a major effort. For example, if instead of using the Domino OSMemoryAllocate() API to obtain shared memory, a program uses the native platform API, then this will have to be ported to each separate architecture/platform, as it does vary greatly.

One other item of complexity to porting between platforms concerns the compilers and flags needed to build Domino. Lotus is working on making this easier by providing a text file as part of the SDK includes which would include all compilation information—OS levels, compiler levels, compilation flags, and linker flags. In the meantime, this information is nestled somewhere within the SDK documentation.

Once again, it can actually be the QE effort that takes the largest amount of time during a port (if the application has been coded to use the Domino OS layer; otherwise, development can take longer). This is due to the fact that the entire application must be tested on the new platform which can take a considerable amount of time, depending on the size of the application.

## 1.4 Structure of Domino on Linux on zSeries

Figure 1-1 on page 7 shows the structure of Domino for Linux on zSeries, either running native on the zSeries server (except for the newest zSeries models), or running in virtual mode under LPAR or z/VM. The numbered items are explained below the table. Then we describe the components in more detail.

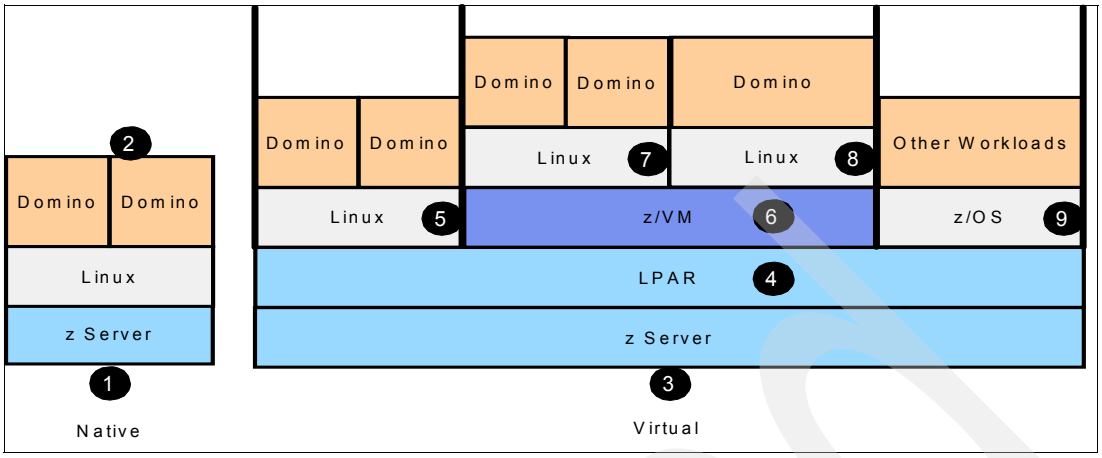



*Figure 1-1 Structure of Domino for Linux on zSeries*

1. Linux running natively on a zSeries server. There is a single instance of Linux, and it is the only operating system on the server. It has access to all of the server resources—processors, memory and I/O. The newer models of zSeries do not support this mode of operation.
2. Multiple Domino servers can be run on the single Linux instance using Domino partitioning (DPAR). We recommend one or two DPARs in a production environment, unless very lightly loaded. As more DPARs are required, add additional Linux copies to run them.
3. Linux running in virtual mode on the zSeries server, under LPAR, or z/VM, or both.
4. LPAR, or logical partitioning, allows multiple operating system images to run on a single zSeries server. The resources are shared between the operating systems that run in logical partitions and are isolated from each other by the server hardware.
5. Linux is running in one of the logical partitions, and has a share of the total server resources. It is supporting two Domino partitioned servers (DPARs), as in the native case.
6. z/VM is running in another logical partition. z/VM allows multiple operating system images to run under it, in virtual machines. These are similar to LPARs, but are more flexible as z/VM is a software implementation. z/VM could also run natively on the server, in which case it would own all the resources, and no other LPARs would be running. VM virtual machines provide isolation from each other, but not to the extent that LPARs do.
7. Linux is running in one of the virtual machines under z/VM. It is supporting two Domino partitioned servers (DPARs).
8. This second Linux instance is also running in one of the virtual machines under z/VM. In this case, it is supporting one Domino server.
9. Other operating systems can run in other logical partitions on the zSeries server. In this case, we show a z/OS operating system supporting other types of workloads on the zSeries server.

Now we describe these components in more detail, to give all readers a deep level of understanding of the characteristics of all of the parts.

### 1.4.1 Domino

Domino is very different from many traditional zSeries applications, and it is worth understanding the key differences.

- A Domino server is the entity that a client connects to. A server has a name, and the files that define the server are the initialization file (notes.ini) and the Domino Directory (names.nsf, previously called the Name and Address Book).
- Each Domino server belongs to a Notes domain. A domain typically contains multiple servers. It is a group of servers that are managed as one, and they all have a copy of the same Domino Directory. For example in IBM, all servers in IBM UK are grouped into the IBMGB domain.
- Multiple Domino servers can run on a single operating system. They are called Domino partitions, often referred to as DPARs. Each Domino server is independent, and they can be in the same or in different Notes domains. You run multiple DPARs if workloads need to be separated (for example, test from production), or if one Domino server is unable to support all of the users. However, as with LPAR, adding more DPARs increases the system overhead, so you should run as few as you can per operating system.
- Domino servers can run on many different platform types, including Windows, Linux, UNIX (IBM AIX, Sun Solaris, HP-UX), iSeries™, and z/OS on zSeries. A single domain can contain servers on different platforms. It is likely that, when installing Domino for Linux on zSeries, the Domino servers will be put into an existing Notes domain with servers on other platforms.
- Domino appears to be much more processor-intensive than traditional zSeries applications. This is because it was developed in the 1990s, rather than the 1970s, when hardware had become much cheaper and software more powerful, so higher functionality was possible. However, this does not equate to high overall costs just based on the processor capacity used. Since Domino is a single application, and requires very little in terms of management software, overall costs are much lower than for a traditional zSeries workload using the same processor capacity.
- Domino uses multiple address spaces to process its work, and some of them run background tasks such as replicating databases between servers, updating database indexes, and compacting databases. Because of these batch tasks, we recommend running Domino on a server with at least two processors, to allow for efficient multitasking.
- Most of the administration of Domino is done by Notes administrators. The zSeries support staff sets up an environment for the Notes administrators in a similar way to setting up an environment for CICS® or DB2. They provide the operating system and install the Domino software, and provide services such as backup and operations, but will probably not be involved in the day-to-day administration of the Notes environment.

### 1.4.2 Linux

Linux is an operating system that comes from a UNIX background. However, unlike the different flavors of UNIX, Windows, and other operating systems, the fact that it is open source software means that it runs on many different types of hardware, including Intel, UNIX servers, and IBM @server iSeries and zSeries.

Linux is attractive for many types of applications, and more and more software vendors are producing products that run on Linux. IBM Lotus Domino is one such product. From a software vendor viewpoint, Linux is very attractive as vendors can develop one version of the software to the Linux interfaces, and then compile it to run on many different hardware platforms.

From a user viewpoint, some of the advantages of Linux are:

- It is more robust than Windows because it comes from a UNIX background.
- It changes the cost model. Although you can download Linux for free from the Internet, you will in reality have many of the same sorts of costs that you have today. You will

probably buy a Linux distribution and a Linux support contract. Software that runs on top of Linux, such as Domino, is still chargeable, as is the server that Linux runs on. However, Linux opens up much more choice and flexibility since it is supported across so much of the IT industry hardware and software, and therefore it enables you to make more cost-effective choices.

- It provides a lot of flexibility. You can choose the hardware platform that best meets your quality of service, cost, and risk needs today, and can easily move to a different hardware platform if your needs change. You can, for example, start with a single Intel server and grow to a single zSeries server without changing the applications.
- Support and security are generally stable since the code is available to, and worked on by, many people around the world. This informal support can be supplemented by a formal support contract from IBM.

### 1.4.3 zSeries hardware

The zSeries hardware traces its origins back to S/360™ (shipped by IBM in 1964), and it has been continuously developed to provide the highest levels of scalability, availability, and manageability for commercial applications; it is the “mainframe” that the industry refers to. Here are some of the key features of the zSeries hardware to be aware of for Domino:

- Linux is supported on all zSeries models—z800, z900 and z990 (which can go up to a 32-way server)—and on older S/390 models with the IEEE floating point hardware feature: G5, G6, and Multiprise® 3000.
- Linux can run native on all of these servers, in which case it has access to all of the server resources. However Linux is not yet proven on high numbers of processors under one operating system instance.
- The maximum number of processors on a zSeries server depends on the model. Advanced design means that zSeries processors can process more commercial work than other types of processors. Techniques include high speed access to memory, caching to get very good performance from symmetric multiprocessors, and dedicated I/O processors that offload work from the central processors.
- zSeries servers include advanced reliability and availability features, including switching to an alternative processor when a production processor fails, advanced memory error correction techniques, and advanced packaging that minimizes the number of components and therefore the failure rate.
- A zSeries processor can contain two types of processors: standard processors can run any support operating system, including Linux, z/VM and z/OS, while Integrated Facility for Linux (IFL) processors can only run Linux, either natively, or in an LPAR, or under z/VM. IFLs have a lower purchase cost than standard processors, and also do not attract licensing charges for software running under z/OS on the same server.
- A zSeries server supports both 31-bit and 64-bit operating systems. Domino for Linux on zSeries runs in 31-bit mode.
- zSeries servers can be upgraded nondisruptively when required; this feature is called Capacity Upgrade on Demand (CUoD).
- zSeries servers can be partitioned, as you see in the next section.

### 1.4.4 zSeries logical partitioning (LPAR)

A zSeries server can be partitioned at the hardware level to support multiple operating systems on one server.

This type of facility is becoming more common on various platforms, but some of the key features of zSeries logical partitioning are:

- Each zSeries server can support up to 30 logical partitions on a z990, or 15 logical partitions on all other zSeries and S/390 processors.
- In each logical partition you can run a different operating system, including Linux, z/VM (which can run any OS, including Linux, in virtual machines), and z/OS.
- Logical partitions can have one or more processors defined to them, and these can be dedicated processors, or shared. Sharing of processors can be done down to less than 1% of a processor. This is managed by the hardware according to partition priorities that the user defines.
- Memory (also known as central and expanded storage) is dedicated to each LPAR, but can be moved between LPARs (subject to the operating system being able to handle that).
- I/O paths can be shared between LPARs, so you do not need as many I/O paths compared to systems that dedicate each I/O path to one partition.
- The partitions are isolated from each other, such that a workload in one partition cannot affect a workload in another.
- zSeries hardware provides a very high speed network connection between LPARs, called HiperSockets™. To the partitions, it looks like an external network connection (thus preserving isolation between partitions), but it is extremely fast since it is implemented within the server.
- The OSA Express card can be shared between LPARs, and allows for a large number of separate network interfaces sharing a single GbE port.

### 1.4.5 z/VM

z/VM is an operating system that runs on zSeries hardware. Like z/OS, it has a long history of providing a highly scalable and reliable operating system environment. z/VM is a relatively “light” operating system, in that it provides limited function. Essentially it runs on the server hardware, natively or in an LPAR, and manages the real hardware resources that are available to it: processors, memory, and I/O. It then provides a virtual machine environment that you can run other operating systems in, including Linux. Some characteristics of z/VM are:

- Thousands of virtual machines can be run at the same time.
- Each virtual machine looks like a real machine; it has a console, card reader (used for input), printer, disks, and tapes. These appear to the guest operating system just like real devices. They may in fact be real devices (for example, a real printer dedicated to one virtual machine), or they may be virtual devices (such as a printer shared between virtual machines, or a printer that automatically sends to the reader of another virtual machines). z/VM handles the mapping between virtual and real devices.
- A unique facility of z/VM is minidisks. z/VM can divide a physical disks into sections, called minidisks, and assign these individually to virtual machines. A guest operating system sees a physical disk volume that has a smaller size than the real disk.
- z/VM provides networking connections between virtual machines, so they can communicate without going outside the server onto the real network.
- z/VM provides prioritization between different virtual machines, so a VM administrator can, for example, set the priority of a production guest to be higher than that of a test guest.
- Additional virtual machines can be started and stopped just by logging onto them or logging them off. New virtual machines can quickly be defined, so this is very flexible.

- A z/VM operator is able to manage the guest operating systems, so you can perform centralized management and operation of multiple Linux guests.
- VM/ESA® 3.1 is also able to support Linux on zSeries.

## 1.5 What's new in Domino 6.5

The most important new feature in Domino 6.5 for readers of this redbook is the addition of support for Linux on the zSeries platform.

Domino 6.5 also includes these features:

- Performance—Platform statistics for Domino on Linux.
- Performance—Expose D6 Workloads (Domino Web Access, Mail, IMAP) in Server.Load.
- z/OS specific—Exploit hardware cryptography for SSL.
- Database unread marks can now replicate between databases on clustered servers or on all servers where the database resides.
- A unified interface for Fault Recovery and Cleanup Scripts. Now administrators can enable and disable NSD to collect diagnostic data, enable/disable server restarts, and specific cleanup scripts, all from within the Domino Server document.
- Storing 10 last (# controlled by notes.ini) versions of server documents in XML format, in the IBM_TECHNICAL_SUPPORT directory, so that administrators can track changes.
- The Automatic Diagnostic Collection tool (ADC) collects diagnostic data (if the server or client crashes) and sends the collected data to a mail-in database when the server or client restarts. You can then use the collected data to determine the cause of the crash.
- Memcheck is now enhanced by locking memory pools when executed, so there are no extraneous errors reported when it does its memory validation.
- Semaphore management now adds time stamps in SEMDEBUG.TXT, to better correlate the time-outs to what is occurring in the server log.

See http://www.lotus.com for more information about Domino 6.5.

2

# Planning

In this chapter, we discuss topics that will help you plan your Domino for Linux on zSeries implementation. These include defining the hardware and software prerequisites, describing the skills needed, and providing planning information for Domino, Linux, and optionally, z/VM.

You will find additional planning information in other chapters, notably in Chapter 4, “Disk configuration” on page 53, and in the installation chapters. It would be useful to read these chapters for planning purposes before you begin your installation.



## 2.1 Building a team

The installation and setup of Domino for Linux on zSeries requires expertise in many areas—Linux, zSeries, Domino, and optionally VM. Specifically, the required skills include:

- Knowledge of your Domino infrastructure, both present and future
- Lotus Notes and Domino administrator skills
- Systems programmer skills in Linux on zSeries
- Systems programmer skills in VM
- Skills in allocating and managing datasets on DASD
- Knowledge of your network and TCP/IP
- Knowledge of your operations environment, including backup and recovery procedures and startup and shutdown procedures

A team approach works best!

Few people have all these skills. For this reason, we strongly recommend using a team approach to the planning and implementation of your Domino for Linux on zSeries system. This will help you to have the appropriate Linux, zSeries, and z/VM skills available. We cannot overstress the need for strong Domino administrator skills, as well.

We recommend that you bring your cross-functional team together right from the beginning, during your initial planning process. During the installation of Linux and Domino, you will be asked to provide several bits of information: the Domino server name, Domino server hostname, Domino domain name, your Organization name, and optionally, a VM guest user ID. If all the members of the team are present when you develop this information, you will have a self-documenting naming convention which will grow with your implementation.

This redbook contains information for each member of your team. We encourage all members to read this chapter and to review other sections of the book as needed. The Administration and Systems Management chapters focus on a specific audience. Chapter 8, “Domino administration” on page 173 is written for the Domino administrator. The topics in Chapter 9, “Systems management” on page 221 are directed toward the system programmer. However, we hope that both Domino administrators and systems programmers will read both chapters.

## 2.2 Deciding where to run Linux on zSeries

The first planning decision to make is where to run Linux on your zSeries system. There are three alternatives.

- Run natively on the zSeries hardware (except on the latest models)
- Run natively in a logical partition (LPAR)
- Run as a guest under VM

The first method could be used when running in a small processor, such as a z800 or Multiprise 3000—but it is not viable in a larger processor, for several reasons: only one Linux system can be run at a time; the Linux server owns all the hardware attached to the server; and Linux does not support all the hardware which can be attached to a zSeries processor. It is not supported in the latest zSeries models. Because of its limited use, we did not consider this any further while writing this redbook.

We do consider the other two methods to be viable, depending on your circumstances; which one you choose depends on your needs. We now describe those two methods in more detail and discuss the advantages of each.

### 2.2.1 Logical partition

A zSeries processor can be partitioned into 15 different logical partitions (LPARs). With the z990, the number of partitions increases to a maximum of 30.

Logical partitions are an allocation of the available processor resource, either shared or dedicated. Processor sharing is possible on a system with any number of CPUs, even if there is only one. The processors can be either standard zSeries processors or Integrated Facility for Linux (IFL) processors.

#### Advantages

You can run up to 30 individual Linux systems in a zSeries processor. Perhaps more compelling, however, is the ability to run Linux LPARs on the same processor with a z/OS or z/VM LPAR. Systems in LPARs are well isolated from each other, from a security perspective.

#### Disadvantages

Linux was not written to maximize usage of resources on zSeries. You will probably be underutilizing the resources that are allocated to an LPAR controlled by Linux.

You must have access to a system console, such as the hardware maintenance console (HMC), to start and restart a Linux partition, or to define or change the definition of a partition. This makes managing your Linux servers more complex and requires the Linux system programmer to work closely with operations.

### 2.2.2 VM guest

Linux can run as a guest in one, or multiple, VM virtual machines. z/VM uses architectural hardware functions in zSeries to virtualize the zSeries instruction set. This means that each guest machine thinks it has its own dedicated zSeries system. A number of guest machines (Linux or others, including z/OS) can run concurrently. The number of concurrent guests is limited only by the resources available to virtualize them.

#### Advantages

The guest console can be accessed by any networking method supported by z/VM, including rlogin, telnet, 3270 through TCP/IP, and 3270 through SNA. This means you can start and restart remotely. Your virtual machine is protected by a user ID and password. You can automate, manage, and administer virtual machines using CMS.

There is also better resource management under VM, because memory and CPU are shared among the Linux guests, and resources are assigned to each Linux guest as it needs them. z/VM was written to optimize resource allocation and utilization among all guests on a zSeries system.

#### Disadvantages

If you want to run multiple Linux systems concurrently, z/VM skills are needed and a number of new concepts need to be learned. Also, VM guest systems are not quite as isolated from a security standpoint as those in separate LPARs.

## 2.3 The effect on Domino

Using the LPAR or VM guest options provide great flexibility in how you structure your Domino for Linux on zSeries system. There are advantages to running either configuration, and there

is no single best approach. To help you determine the right configuration for you, we now discuss how the configuration decision affects Domino.

### 2.3.1 LPAR

You can run several Domino servers in a single Linux for zSeries LPAR. These can be partitioned servers or (with Domino 6) independent servers. Partitioned servers share the same executable Domino code. Independent servers have separate program directories and executables. Domino 6 introduced the ability to run more than one independent Domino server in an LPAR. This provides you the flexibility to manage the servers independently, even though they are running in the same LPAR.

#### Advantages

When running multiple partitioned servers in an LPAR software, upgrades are simpler. You can upgrade all servers at one time. You can assign several CPUs to an LPAR, thus allowing you to scale to larger Domino servers.

#### Disadvantages

The Linux that supports Domino on zSeries is a 31-bit implementation, so the LPAR is effectively limited to 2 GB of memory. Although you can use XPRAM for paging to expanded memory, it is not clear that this significantly improves the scalability of Domino.

If you run independent servers, they must be upgraded individually. Although offering more flexibility, this adds to the administrative workload.

### 2.3.2 VM

With z/VM, you can run multiple Linux guests in a single z/VM LPAR. This gives you more options. You can run each Domino server in its own Linux guest. You can also run multiple Domino servers in a single Linux guest, just as you can run multiple Domino servers in an LPAR without VM.

#### Advantages

If each Domino server is in a separate VM guest, you have better server isolation. You can easily separate servers for development and testing from your production servers. You can also separate application servers from mail servers. It is an ideal environment for an application service provider (ASP) who wants to run many small Domino servers for separate customers. z/VM lets you set the execution priority of individual guests so you can, for example, provide a higher level of service to a production mail server and lower priority to an application development server.

With z/VM, you can utilize the memory above 2 GB (which Linux in 31-bit mode and Domino cannot), which can be useful if you are running a number of Linux guests. You also gain a single point for administering all of your Linux installations. z/VM does a good accounting job for resources used (unlike a Linux LPAR) and allows you to charge Linux systems accurately. z/ VM is written to maximize utilization of zSeries resources.

#### Disadvantages

There is some overhead in having z/VM. You would probably not select this method if you are planning to stay with one or two Domino servers. However, VM does manage and utilize zSeries resources better than native Linux.

### 2.3.3 Conclusion

Ultimately, the best way to run Domino for Linux on zSeries depends upon your implementation. If you are planning just a few Domino servers and do not already have z/VM, you may have no reason to bring it in; Linux will run Domino servers in a native zSeries LPAR. You should monitor utilization of resources such as memory to ensure that Linux is maximizing the zSeries potential.

On the other hand, if you are planning to have more than a few Domino servers, even if some of the servers are small, then we recommend that you implement them under VM.

## 2.4 A few large servers - or many small servers

Is it better to run a few large Domino servers or many small Domino servers? Again, the answer is: it depends. Consider the function of the Domino server as well as the capabilities of the hardware and software before making your decision. Here are some general rules (which have exceptions).

- In the case of mail servers, it is better to consolidate users into fewer Domino servers to keep mail transfer between servers to a minimum. It is more expensive and takes more time to transfer mail to recipients on another server than it does to deliver the mail locally.

  If you are moving small mail servers from another platform to zSeries, you should consider consolidating the users into fewer mail servers, and not simply transplanting the small servers.

  On the other hand, you probably will not place all your users on a single Domino mail server, even if the zSeries hardware makes that possible. The larger the Domino server, the more data there is and the more critical it is to keep the server available to the users. That makes it more difficult to find a window for backup and maintenance. In addition, VM is good at optimizing resources for lots of smaller servers.
- When using application servers, you may choose to have several smaller servers with the same application database and use replication to keep the databases synchronized. This is especially useful if you have users accessing the database from around the world. You can then take one Domino server offline for maintenance without impacting other users.

  If you are moving small application servers to Linux on zSeries, it makes sense to move them intact and keep their identity. In this case, running them under VM is a good choice.
- z/VM makes it easy to support many small Domino servers. With z/VM you could easily transplant your development and test servers to a single hardware server while keeping the Domino server infrastructure.

You will find that it is a matter of balancing the advantages of managing a few servers with the flexibility of managing many servers. You may find that you will have a few large mail servers and many small application and development servers.

## 2.5 Required hardware and software

The following hardware and software is required for Lotus Domino for Linux on zSeries.

### 2.5.1 Hardware

Domino for Linux on zSeries requires the following hardware.

## Processor

- A processor capable of running Linux for zSeries.

  Linux for zSeries requires IEEE floating point hardware feature; therefore the minimum hardware server is a S/390 model G5 or G6, a Multiprise 3000, or a zSeries model z800, z890, z900, or z990.

  If running under VM, the processor must be capable of running VM/ESA V3.1 or z/VM V4.3 or higher. The processors may be standard zSeries processors or Integrated Facility for Linux processors (IFLs).

  For testing and limited production, a single processor is sufficient. However, Domino has many tasks and runs better in a multiprocessor environment, and the minimum recommended configuration for full-scale production is two processors or IFLs. The processors can be shared with other LPARs but, depending on the size of your Domino system, you may need additional processors.

- The minimum memory requirement is 128 MB. With 128 MB you can bring up a Domino server with a couple of users and do some testing.

  The amount of memory you will need for a production server depends on the number of users and applications. We recommend a minimum range of 256 to 512 MB for a pilot (if you run under VM, we recommend a minimum of 512 MB). We provide further guidance on this subject in Chapter 11, “Capacity planning for Linux on zSeries” on page 273.

  Linux on zSeries (31-bit mode) supports up to 2 Gigabytes of memory. Domino servers supporting hundreds of users will typically require more than 2 Gigabytes of memory. When configuring Domino for Linux on zSeries in LPAR mode for a large number of users, XPRAM can be considered for paging to fast memory rather than to disk. See the Domino Release Notes for information about setting up and using XPRAM. If your Linux runs under VM, then swap to a virtual disk instead of XPRAM.

- TCP/IP networking support.

## DASD

You will need a minimum of three volumes of DASD space for the initial installation of Domino; we recommend four or more volumes if you are installing a mail server. Table 2-1 describes how those volumes are used.

In addition to the DASD needed for Domino, you will need to consider the Linux swap space and VM page space. For more discussion on these topics, refer to the chapters on installation, performance, and capacity planning.

*Table 2-1 DASD Volumes and recommended minimum size*

| Partition | Description | Size |
|---|---|---|
| / | Root partition | One 3390-3 |
| /opt/lotus | Product files | One 3390-3 (see Note 1) |
| /notesdata | Domino control files, databases, and templates | One 3390-3 (see Note 2) |
| /notesdata/mail | Notes mail files | One 3390-3 (see Note 2) |

**Note 1:** The Domino executables and .tar file will consume approximately 65% of a 3390-3. After installation of Domino, you can remove the .tar file and recover approximately 600 MB, however, you will need space to store a new .tar file when you upgrade Domino. One 3390-3 is sufficient for the mountpoint of /opt if you do not have other products to install.

**Note 2:** The amount of disk space that you will use for Notes databases and mail files depends entirely on the size of your planned installation. We recommend that you size those requirements carefully and design your filesystem and directory structure accordingly.

Refer to Chapter 4, “Disk configuration” on page 53 for more information and recommendations about filesystem layout. This chapter contains topics like 4.4, “Logical Volume Manager (LVM)” on page 61, 4.5, “Domino filesystem structure” on page 66, and 4.6, “Placement of high-use Domino files” on page 68.

### Workstation

- You will need a workstation with a CD-ROM drive and a network connection to the zSeries where Domino will be installed. Domino for Linux is shipped on CD-ROM, so this workstation will be used to transfer the code to the zSeries.

  FTP is the recommended file transfer method. You may either enable the FTP server on your Linux system or, if that is not available, configure your workstation to run an FTP server. In the latter case, you log on from the Linux server using an FTP client and get the TAR file from the CD.

- You should have the Domino Administrator client with the Remote Server Setup feature installed on the workstation from which you will perform the Domino server setup. This can be either a Domino 6.0 or Domino 6.5 Administrator client. All the Lotus Notes clients are available on the Lotus Notes CD, which is separate from the Domino CD. We discuss this in 7.5, “Setting up the Domino server” on page 152.

  **Note:** This step can be done any time; you do not have to wait until you are in the middle of installing a Domino server to install the client on a workstation.

## 2.5.2 Software

Lotus Domino for Linux on zSeries requires the following software:

- United Linux v1.0 for IBM S/390 with Service Pack 2 or Service Pack 3. This is equivalent to SuSE Linux Enterprise Server 8 (SLES 8) for IBM Mainframes. Service Pack 3 is now the recommended level.

  The Linux code is acquired from SuSE, not IBM. For more information, refer to the SuSE Web site:

  http://www.suse.de

- If running under VM, you will need either:
  - VM/ESA Version 3.1
  - z/VM Version 4.3 or higher.

  While writing this redbook, we were running under z/VM Version 4.3.

### Workstation

To install the Domino software, you need the following clients.

#### *Installation client*

You will need either a telnet or ssh client to install the Domino server. If telnet is not available in your installation, you can use one of the ssh clients, like PuTTY. There are a number of free secure shell clients available. We used PuTTY, a free download from:

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

You can also find more information about ssh at:

http://www.openssh.com/

### *Domino Administrator client*

You must have a Domino Administrator client to administer your Domino servers. We recommend you use the Administrator client to set up your Domino for Linux on zSeries server using the Remote Server Setup function within the client. The Remote Server Setup function is an optional feature when you install your Domino Administrator client from the Lotus Notes CD (a separate CD from the Domino CD). If you install all clients and features, you will have the Remote Server Setup option available to you. After installation, the application is *serversetup.exe* in the Notes directory. Or you can call it from the Windows start menu, as shown in Figure 2-1.


*Figure 2-1 Start Menu - Remote server setup*

## 2.6 Linux planning and customizing

Domino for Linux on zSeries requires United Linux 1.0 with Service Pack 2 or 3. You can see if you have the right level of Linux with this command.

```
uname -a
```

The result should be:

```
linuxb:~ # uname -a

Linux linuxb 2.4.19-4suse-SMP #1 SMP Thu Jun 5 23:01:37 UTC 2003 s390 unknown
or
linux linuxb 2.4.21-107suse-SMP #1 SMP Thu Mar11 17:20:12 UTC 2004 s390 unknown
```

The date on your system must be later; the dates above are the minimum dates.

The sys_epoll function, introduced in Service Pack 2 of United Linux 1.0 for S/390 or SLES 8, is required to run Domino. To be sure that epoll has been installed into the kernel, issue this command.

```
grep epoll /boot/System.map*
```

The result should be:

```
>grep epoll /boot/System.map*
000797d8 T sys_epoll_create
00079854 T sys_epoll_ctl
000799b8 T sys_epoll_wait
```

### 2.6.1 Linux kernel changes

You will also need to change some Linux kernel values to optimize Linux for Domino use. Refer to 7.3, “Pre-installation steps” on page 135 for details.

### 2.6.2 Linux sysstat package

Domino can collect platform statistics for display on the Domino Administrator client. Some of these (for example, the logical disk statistics) depend on the **`iostat`** command for information. If **`iostat`** is not available to Domino, the disk platform statistics will not be available. **`iostat`** is typically delivered in the sysstat package on Linux.

**Note:** Taking the defaults did not install the package on our SuSE SLES8 system.

### 2.6.3 Linux user IDs

To install Domino on your Linux system, you need at least two Linux user IDs:

- The root (UID = 0) for extracting the TAR file and installing the binaries.
- One ID (UID ≠ 0) under which the Domino server will run. During the installation process, this ID is needed to set up and configure the server.

  You need one user ID for each Domino server running in the Linux server.

You may also want to define a user ID that will be used to FTP the Domino code from the workstation to the Linux server, because root is normally not allowed to use FTP.

You will also need a group to which the Domino server belongs. This group must be defined before installing the Domino server. The default group, which may be changed during installation, is *notes*.

### 2.6.4 Linux network

For each Domino server you plan to install, you need one complete IP address. That means the combination IP address and port (1352 for Notes RPC).

**Note:** We do not recommend the use of Domino port mapping. Although that makes it possible to have multiple Domino servers using the same IP address on different ports, it is not advisable because the port mapping function is performed by one of the servers—and if that server is not available, clients cannot reach the other servers even if they are running.

For installation, you need FTP and telnet or ssh access to your Linux system. Make sure the corresponding services are available.

## 2.7 VM planning considerations

Refer to Chapter 3, “z/VM basics, planning, and tasks” on page 29 for more information about z/VM. In this section, we list some basics to keep in mind while planning your environment to exploit z/VM.

### 2.7.1 VM user IDs

You will need one VM user ID for each Linux guest. Example 2-1 on page 22 is a sample definition with minimal DASD defined for initial installation.

Depending on the size of the Domino server you are implementing, you will need to adjust the DASD and memory.

*Example 2-1 Sample VM user definition*

```
USER LINUXB AAABBB5G 512M 1G G
IPL CMS PARM AUTOCR
MACHINE ESA
DEDICATE 2C08 2C08
DEDICATE 2C09 2C09
DEDICATE 2C0A 2C0A
MDISK 0191 3390 1545 50 VMLU1A MR
MDISK 0200 3390 0001 3338 LX1518 MR
MDISK 0201 3390 0001 3338 LX1558 MR
MDISK 0202 3390 0001 3338 LX1598 MR
MDISK 0203 3390 0001 3338 LX15D8 MR
```

## 2.7.2 Minidisks

VM minidisks are virtual disk devices. They are implemented by partitioning a real volume into cylinder ranges that appear as separate disk volumes to the virtual machine. A minidisk can span an entire real disk volume if you desire. The Linux user account of the Domino server will have a 191 minidisk. The number of minidisks the Domino server will need for the Domino data directory and mail and application directories depends on the size of your planned installation.

Minidisks can be shared or non-shared. If authorized, one virtual machine can link to a minidisk belonging to another virtual machine to access the data on it. Links can be either read-only or read-write. When a minidisk is write-shared, some software is needed to manage access to the data. However, Domino data cannot be shared with other Domino servers or users.

### Virtual minidisks

Virtual minidisks have similarities to temporary minidisks. Instead of being mapped to cylinders of real disk volumes, they are mapped into real storage by CP. This means that they avoid the need for disk I/O. The associated pages are managed by CP as part of its overall real memory management.

The amount of DASD you configure in your VM system depends mainly on the requirements of your VM guests and your Domino infrastructure.

## 2.7.3 Page and spool space

z/VM requires adequate space to perform well when more virtual memory is defined than physical memory. A rule of thumb is to have at least twice as much page space on DASD as the sum of your total virtual storage and virtual disk. Ensure you have defined sufficient page space defined for all your Domino servers.

## 2.7.4 Memory

Each virtual machine has its own defined virtual storage or memory. CP manages the residency of virtual machine pages in real storage with a sophisticated paging mechanism. Pages that have not been referenced can be moved out of real storage, either into expanded storage or onto a paging device. When a virtual machine touches a page that is no longer in real storage, a page fault occurs and CP will bring the missing virtual page back into real storage.

A portion of the real storage on a zSeries system can be dedicated to a virtual machine. In this case, the storage of the virtual machine is real and the operating system running in the virtual machine can perform its own memory management without intervention by CP. Expanded storage can also be dedicated to a virtual machine.

CP also has facilities that allow the sharing of virtual pages by a number of virtual machines. A shared virtual page requires just one page of real storage no matter how many virtual machines are sharing it, thereby economizing on real storage requirements. Domino does not take advantage of this sharing. Domino partitioned servers share executables using facilities within Domino.

The amount of memory that z/VM has is very important to the performance of almost any application. zSeries memory can be allocated as either central or expanded. A rule of thumb is to use a 3:1 ratio of expanded to central storage. Refer to Chapter 11, “Capacity planning for Linux on zSeries” on page 273 for help in determining how much memory you need in your installation.

### 2.7.5 CPUs

The number of processors that z/VM has is also important. Processors on z/VM can be physical or logical. It is possible to define more logical CPUs than there are physical CPUs; however, this will generally not improve performance, and it is not recommended.

CPUs can either be dedicated to an LPAR, or shared across different LPARs. A minimum of two physical CPUs is recommended, as Domino performs better in a multiprocessor environment. Dedicated CPUs are recommended if you have only two, though shared CPUs are an option for smaller loads.

## 2.8 Domino planning

A successful production environment requires strong Domino administration skills. You should not attempt to implement Domino for Linux on zSeries without these skills. You can obtain them by hiring an experienced Domino administrator or contracting with a services organization.

If this is your first installation of Domino, there are many things to plan, including the following:

- Defining an organizational structure including domain and certifiers
- Setting up the Domino server infrastructure
- Defining user and server naming conventions
- Choosing a pilot application
- Planning a replication strategy
- Defining a mail routing topology
- Planning the Domino directory structure

A comprehensive discussion of these topics is beyond the scope of this redbook. There are, however, several other good sources of information about this subject that you can refer to:

- Other IBM Redbooks; see “Related publications” on page 397.
- Lotus yellow books.
- Lotus Domino documentation, including the Lotus Domino Administrator 6 Help database.

- Classes that are available from IBM and Lotus business partners. Refer to Appendix C, “Educational material” on page 383 for a list of recommended courses.

### 2.8.1 zSeries - just another server platform

Linux on zSeries is just another server platform to Domino. The Notes architecture means that the server platform is transparent to the users.

- Users access a Domino for Linux on zSeries server in exactly the same way that they access any other Domino server. It looks the same, and users cannot tell what the underlying platform is. Domino for Linux on zSeries supports all the clients that are supported on other platforms including Notes, Domino Web Access (formerly known as iNotes™), POP3, and IMAP.
- A Domino administrator can manage a Domino server on Linux using the same tools they use to manage any other Domino server, including the Domino Administrator client or the Domino Console. The only difference between Domino on zSeries and other platforms is that there is no native console function on the zSeries processor. This results in some minor operational differences. For example, you cannot start a Domino server from the Domino Administrator client, although you *can* start it from the Domino Console. For more information about administering Domino on Linux, see the chapters on Domino Administration and Systems Management.

This means that there is very little Domino planning that is unique to Linux on zSeries. If you are moving servers from Windows platform, however, there are a few differences.

- Domino on Linux is case-sensitive.
- Domino on Linux requires TCP/IP for networking.
- Some applications, such as fax or telephony, won’t run on zSeries because the hardware support is not available.

For more information about migrating from another platform, refer to Chapter 14, “Migrating to Domino 6.5 for Linux on zSeries” on page 357.

Table 2-2 lists the information you should gather before beginning the installation, as well as the values we used for our server during the writing of this redbook.

*Table 2-2 Our Domino environment*

| Field | Default value | Settings used in this redbook |
|---|---|---|
| Domino server name | none | domservb/ITSO |
| Domino domain | none | ITSO |
| Domino certifier | none | ITSO |
| Linux user ID for Domino server | notes | domservb |
| Linux group for Domino server | notes | notes |
| Install directory for Domino code | /opt/lotus | /opt/lotus |
| Notes Data Directory | /local/notesdata | /domservb/notesdata |
| Notes Mail Directory | /local/notesdata/mail | /domservb/notesdata/mail1<br>/domservb/notesdata/mail2<br>/domservb/notesdata/mailn |
| VM user ID for Linux guest | none | LINUXB |

| Field | Default value | Settings used in this redbook |
|---|---|---|
| DNS domain | none | itso.company.com |
| Domino server hostname | none | domservb.itso.company.com |

### 2.8.2 Server consolidation

There is one area of planning which is unique to Domino on zSeries: choosing the number of Domino servers to run. Moving to Linux on zSeries provides the opportunity to examine your Domino infrastructure to see if there are benefits to server consolidation. In zSeries there are two forms that server consolidation can take.

1. Consolidate the existing Domino servers to a single hardware server, while retaining your logical Domino server infrastructure. If you have 15 Domino servers today, you will have 15 Domino servers on zSeries.
2. Consolidate the number of Domino servers. Move your servers to the new platform and reduce the number of Domino servers by exploiting the scalability of the zSeries platform.

Option 1 is done most easily when running under z/VM. Option 2 can be done under z/VM, or in a native LPAR. Which of these consolidation plans you implement depends in part on the function of the Domino servers. It is a good idea to consolidate mail servers, because fewer mail servers are more efficient than many; refer to our discussion in 2.4, “A few large servers - or many small servers” on page 17.

In addition to the Domino planning considerations, you may also be consolidating Domino servers that are running on other platforms, such as Windows or Linux on Intel. When implementing Domino for Linux on zSeries, you have an opportunity for different kinds of server consolidation; refer to 14.2.3, “Server consolidation tips and methods” on page 360 for a detailed discussion.

### 2.8.3 Partitioned servers

You can run multiple Domino servers within a single zSeries LPAR or VM guest. Prior to Domino 6, these servers would all be partitioned servers, and they would be running the same version of Domino. Domino 6 introduced the ability to run multiple versions of Domino in a single OS image on UNIX systems. These can be either partitioned servers or independent servers.

- With Domino partitioned servers, all partitions share the same Domino program directory, and thus share the set of Domino executable files. However, each partition (Domino server) has its own notesdata directory and NOTES.INI file; thus each has its own copy of the Domino Directory and other administrative databases.
- With non-partitioned or independent Domino servers. each Domino server has its own program directory; code is not shared between the Domino servers. Each Domino server also has its own notesdata directory and NOTES.INI file. To run different levels of the Domino code requires that you install Domino multiple times, installing the executables at different directory locations. For example,

  At /opt/Dom650/lotus - Domino 6.5 code

  At /opt/Dom651/lotus - Domino 6.5.1 code

  For each of these versions of Domino executable code, you can have partitioned servers. They share the executable Domino code, but each partitioned server has its own notesdata directory.

### Deciding whether to use partitioned servers

Implementing partitioned servers is most useful when all servers have the same function and are managed together. Since the servers share the same set of executables, you upgrade all the servers at one time. This reduces the management of multiple servers. If you plan to have multiple mail servers in an LPAR, these are good candidates for partitioning.

On the other hand, if you have one mail server and one application server, you may want to keep them as independent servers so that you can manage them independently. Although you must upgrade them individually, you have the flexibility to run them at different Domino releases. You can upgrade one server more frequently than another. You may also make changes to one server while the other server is running.

When you install or upgrade Domino for Linux, you will be given the opportunity to install a new Domino server or to add data directories only. The first option will install the Domino executables and create or upgrade the templates and system databases in the Domino data directory of your Domino server. You can be installing or upgrading an independent server or partitioned servers. If you choose to add data directories only, the install program will create a new Domino data directory and tie it to the executable code you have previously installed. In other words, you will create or add a partitioned server. When you select add data directories only, the executables are not changed.

In Chapter 7, “Domino installation” on page 133, we first describe the installation of a single Domino server, and then describe how you would install multiple partitioned servers.

## 2.9 Where to find more information

- “Related publications” on page 397 contains a list of IBM Redbooks about Domino, Linux on zSeries, z/VM, and other topics that might be useful to you.
- Official Lotus product documentation in the Lotus Documentation Library is available online:

  http://www.lotus.com/ldd/doc

  From there you can download the following resources.

  - The Release Notes for specific releases of Domino. These highlight new enhancements and troubleshooting tips, and provide updates to the product documentation. Read the Release Notes for your level of Domino before you begin the installation. The Release Notes for Domino 6.5 include a section specific to installation on Linux in a zSeries environment.
  - Lotus Domino Administrator 6.5 Help, a Notes 6 database which describes how to set up and administer Domino servers. This is the same help file which is installed with the Domino Administrator client. You will also find a copy of this database on the Domino for Linux installation CD. This is a primary resource for Domino administrators who want to learn how to do something with a Domino server.
  - The Lotus product manuals (yellow books) in PDF format. This includes such books as *Installing Domino Servers, Administering the Domino System,* and *Administering Domino Clusters*. These books can also be purchased in hardcopy from the IBM Publications Center, and are available in downloadable PDF format from the Lotus Documentation Library.
- The Lotus Developer Domain is a good source of information about Domino:

  http://www.lotus.com/ldd

  In addition to the Documentation Library, there are technical articles in “LDD Today”, and forums where you can participate in discussions with other Domino users.

- Tutorial: *Introduction to Administering IBM Lotus Domino 6/6.5: Operating Fundamentals*

  If you are new to Domino administration, here is a self-paced, e-learning tutorial to help you get started. It introduces the basic concepts that provide the foundation for Lotus Domino and Lotus Notes.

  http://www-1.ibm.com/support/docview.wss?rs=0&uid=swg27005591

- zSeries has a Web site with information about Domino running on zSeries servers. Here you will find information specific to Domino and zSeries, including news and information, customer case studies, upcoming events, recommended service levels, and capacity planning. The site is currently contains information about z/OS primarily, but Linux information will be added.

  http://www.ibm.com/servers/eserver/zseries/software/domino/

3

# z/VM basics, planning, and tasks

In this chapter, we discuss:

- z/VM basics
- z/VM planning
- z/VM tasks

## 3.1 z/VM basics

In this project, we used z/VM to provide multiple Linux environments on the same zSeries system.

Figure 3-1 shows a highly simplified and stylized view of a zSeries processor and its physical components or resources—processor units, memory (real and expanded storage), channels, control units, and devices. All of these are managed by an operating system; typically this is z/OS, z/VM, or Linux.

The original Virtual Machine (VM) operating system was developed in the late 1960s as a tool to aid in development and testing of new operating systems and software without the need to tie up an entire physical computer and associated hardware. It did this by allowing the creation of multiple virtual computers along with all hardware resources under the VM operating system. z/VM still works this way today. The resources that make up these virtual machines are defined by the directory entries of the user ID under which the virtual machine will log on. These virtual machines appear to the operating systems that run under them as real computers with real hardware.

z/VM has special abilities, however. Its two most important components are the control program (CP) and the Conversational Monitor System (CMS). CP is able to *virtualize* hardware resources. It does this either by partitioning or sharing real hardware resources, or by emulating their behavior programatically.

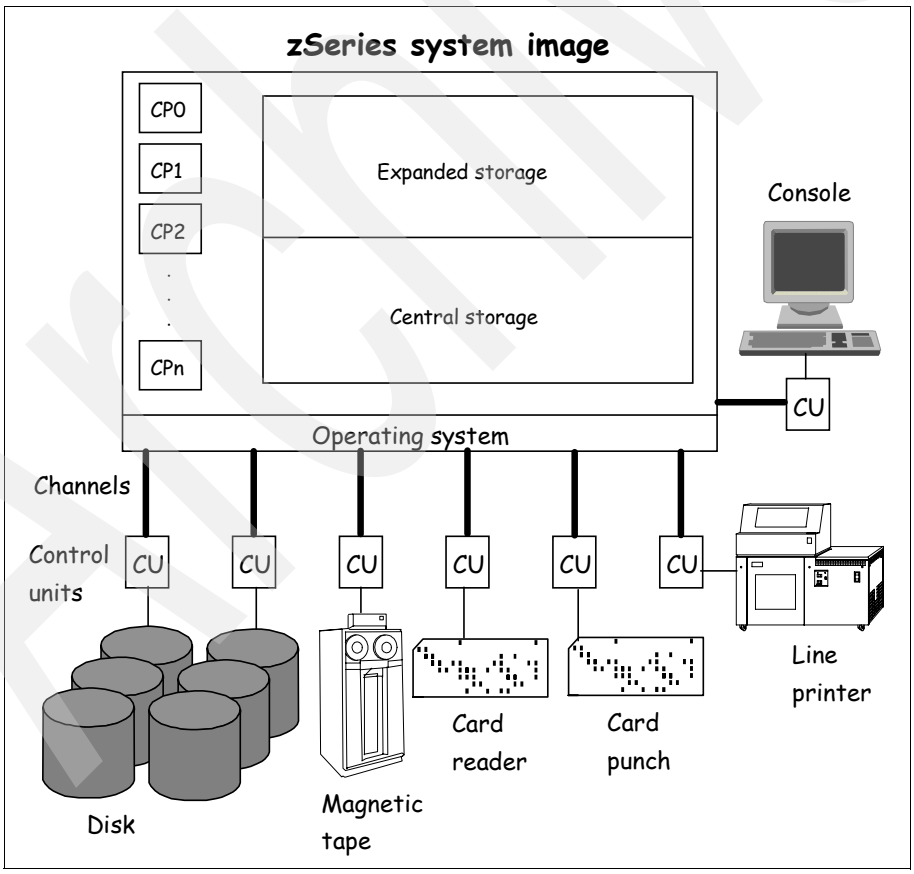



*Figure 3-1 zSeries system resources*

CP implements z/Architecture™ or S/390 architecture virtual machines. On a single piece of hardware, you can run several copies of the same or different operating systems under the

control of CP. Any operating system that can run on S/390 or zSeries hardware can run under z/VM as a guest operating system in the z/VM-created virtual machine.

From the point of view of the operating system running inside the virtual machine, the hardware appears to be real hardware. These running operating systems are sometimes called *guests*. They can be added or deleted at will by changing the *CP directory.* We discuss this later.

We refer to z/VM as a *hypervisor*. The users of each virtual zSeries machine are unaware that a hypervisor is providing the zSeries environment in which their application is running. Each virtual machine has its own virtual memory, virtual devices, virtual processors, and other virtual resources.

Within each virtual machine, a zSeries operating system is IPLed. You can even IPL z/VM within a virtual machine. Each operating system running in its own virtual zSeries environment communicates with virtual devices. The mapping of virtual to real devices and resources is handled transparently by CP. The result is that by running z/VM on the processor shown in Figure 3-1 on page 30, we can replicate the zSeries environment many times over. This is illustrated in Figure 3-2 on page 34.

## The CP or user directory

The definition of a virtual machine is stored in a master directory called the *CP directory* and also often called the *user directory*. For example, the source listing of a user directory entry for one of our Linux guests is as follows:

*Example 3-1 Directory entry for Linux guest*

```
USER LINUXC XXXXXXXX 512M 1G G 1
IPL CMS PARM AUTOCR 2
MACHINE ESA 4 3
DEDICATE 2C0C 2C0C 4
DEDICATE 2C0D 2C0D 4
DEDICATE 2C0E 2C0E 4
DEDICATE 2C60 2C60 5
DEDICATE 2C61 2C61 5
DEDICATE 2C62 2C62 5
CONSOLE 0009 3215 6
SPECIAL 0600 HIPER 3 SYSTEM GUESTLAN 7
SPOOL 000C 3505 A 8
SPOOL 000D 3525 A 8
SPOOL 000E 1403 A 8
LINK MAINT 0190 0190 RR 9
LINK MAINT 019E 019E RR 9
LINK MAINT 019F 019F RR 9
LINK MAINT 019D 019D RR 9
MDISK 0191 3390 1595 50 VMLU1A MR 10
MDISK 0200 3390 0001 3338 LX1519 MR 10
MDISK 0201 3390 0001 3338 LX1559 MR 10
MDISK 0202 3390 0001 3338 LX1599 MR 10
MDISK 0203 3390 0001 3338 LX15D9 MR 10
MDISK 0204 3390 0001 3338 LX8026 MR 10
MDISK 0205 3390 0001 3338 LX8027 MR 10
MDISK 0206 3390 0001 3338 LX8125 MR 10
MDISK 0207 3390 0001 3338 LX8126 MR 10
MDISK 0208 3390 0001 3338 LX8127 MR 10
MDISK 0209 3390 0001 3338 LX8225 MR 10
MDISK 020A 3390 0001 3338 LX8226 MR 10
MDISK 020B 3390 0001 3338 LX8227 MR 10
MDISK 020C 3390 0001 3338 LX151E MR 10
```

```
MDISK 020D 3390 0001 3338 LX151F MR 10
MDISK 020E 3390 0001 3338 LX155E MR 10
MDISK 020F 3390 0001 3338 LX155F MR 10
MDISK 0210 3390 0001 3338 LX159E MR 10
MDISK 0211 3390 0001 3338 LX159F MR 10
MDISK 0212 3390 0001 3338 LX15DE MR 10
MDISK 0213 3390 0001 3338 LX15DF MR 10
MDISK 0301 FB-512 V-DISK 204800 MR 11
MDISK 0302 FB-512 V-DISK 204800 MR 11
MDISK 0303 FB-512 V-DISK 204800 MR 11
MDISK 0304 FB-512 V-DISK 204800 MR 11
MDISK 0305 FB-512 V-DISK 204800 MR 11
MDISK 0306 FB-512 V-DISK 204800 MR 11
MDISK 0307 FB-512 V-DISK 204800 MR 11
MDISK 0308 FB-512 V-DISK 204800 MR 11
MDISK 0309 FB-512 V-DISK 204800 MR 11
MDISK 0310 FB-512 V-DISK 204800 MR 11
SPECIAL 808 CTCA TCPIP12
SPECIAL 809 CTCA TCPIP12
```

Most of the statements in this entry define virtual resources or devices.

**Notes:**

1 The user ID that identifies this virtual machine is LINUXC. The password field shows XXXXXXXX, but this is DirMaint™ hiding the true password. In the USER DIRECT file without DirMaint, the real password will be in this field.

The virtual machine is defined with a default storage of 512 megabytes, but this can be redefined up to a maximum of 1 gigabyte. This is probably a good starting point for a server that will run Domino. Domino test or development servers may run well with 256 MB.

2 This statement indicates what operating system will be IPLed when you log on to the virtual machine. In our example, an IPL of CMS will occur. We did this because we needed to set up some changes to the normal environment, as well as run some REXX EXECs to set up Linux to run Domino more effectively.

This was done through the user ID's PROFILE EXEC. If no changes need to be made, you can IPL Linux from the minidisk at address 0200 at logon time by changing the directory statement to:

```
IPL 200
```

3 This statement describes the processor architecture of the virtual machine. The maximum number of processors that can be defined for this virtual machine is four. The default is one.

4 Dedicate statements for the OSA Express address triplet. Three are required one for input control, one for output control, and one for data. These three addresses are associated with a single IP address.

5 When running a Domino server, you might need a second IP address. This may be used for clustering or a second partition in the server. If this is the case, a second set of dedicate statements for the address of the second OSA adapter will be needed. Three are required: one for input control, one for output control, and one for data.

These three addresses are associated with a single IP address. This second OSA will need to be on a different physical adapter than the first and have a different MAC address

6 The CONSOLE statement defines the operating console for the virtual machine.

**7** These statements define the HiperSocket address triplet for use with the Guest LAN named GUESTLAN.

**8** These statements define a virtual reader, punch, and printer for the virtual machine.

**9** These are read-only links to minidisks owned by other virtual machines. They typically provide services to the virtual machine. They are sometimes referred to as “system disks”.

**10** These statements define twenty-one minidisks owned by the LINUXC user ID

The general format of the minidisk statement is:

```
MDISK aaaa tttt ffff ssss vvvv rr
```

where:

`aaaa` = the virtual address of the minidisk. This the address that Linux will use
`tttt` = the disk type of the physical disk, typically 3390 for physical DASD
`ffff` = the starting location of the first cylinder of the minidisk
`ssss` = the minidisk size in cylinders (where a cylinder is about 700KB after formatting)
`vvvvv` = the VolSer of the physical DASD
`rr` = mode

For example, consider the following MDISK statement:

```
MDISK 0191 3390 1595 50 VMLU1A MR
```

Device 191 is a minidisk (partition) of Volume VMLU1A that begins at cylinder 1595 with a size of 50 cylinders. It occupies the following cylinder ranges from 1595 to 1645 on the device. This is the Linux guest virtual machine’s CMS A disk.

Device numbers 200-213 are minidisks on 14 different volumes.

Device number 191 occupies cylinder range 1595-1645 on the real disk with VOLID VMLU1A. Device number 200 occupies all cylinders 1-3338 on the real disk with VOLID LX1591. Cylinder 0 is typically used by the z/VM operating system and usually is not assigned to a minidisk.

A Domino for Linux guest might use device 200 for /root, device 201for /opt/lotus, and place devices 202-213 in a Linux logical volume to hold the notesdata directory and mail and application databases.

These are all read/write minidisks.

**11** These statements define the virtual minidisks. We used these primarily for Linux swap space.

The format is of the statement is:

```
MDISK 0310 FB-512 V-DISK 204800 MR
```

This MDISK statement defines 204800 fixed 512-byte blocks (about 100 MB) in read/write mode and assigns it an address of 310. In all, 10 minidisks are assigned with addresses from 301-311. These addresses were used for /swap and for the Domino view rebuild directory.

**12** These statements define the read/write pair of a virtual channel-to-channel adapter.

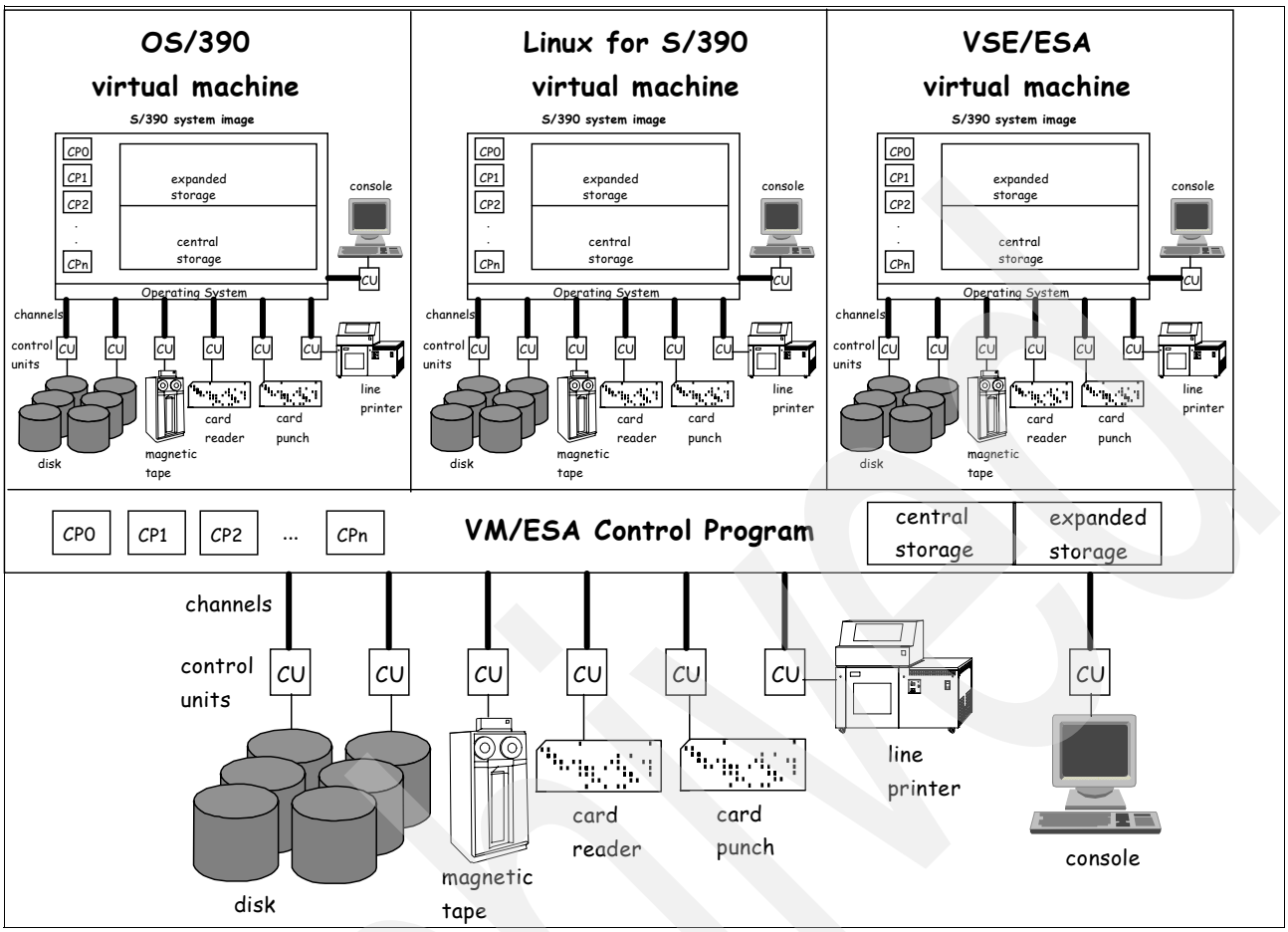



*Figure 3-2 Virtual machines running under the control of a hypervisor*

Now we take a brief look at how CP implements certain device types. Because the S/390 virtual machine concept dates back to the late 1960s and early 1970s, some of the standard devices such as a card reader and card punch may seem antiquated, but they are still extremely useful when virtualized.

## Processors

A virtual machine can have up to 64 virtual processors defined (on 64-bit hardware). If the operating system running in the virtual machine is multi-processor (MP)-capable, it will dispatch work on its virtual processors as though it were running without a hypervisor (on the real hardware). CP will handle the dispatching of virtual processors on the real processors available to that virtual machine. A real processor can be either dedicated to a virtual machine, or shared among virtual machines. Dedicated processors can only be used by *one* virtual machine.

## Storage

Each virtual machine has its own defined virtual storage or memory. CP manages the residency of virtual machines' pages in real storage with a sophisticated paging mechanism. Pages that have not been referenced can be moved out of real storage, either into expanded storage, or onto a paging device. When a virtual machine touches a page that is no longer in real storage, a page fault occurs and CP will bring the missing virtual page back into real storage.

A portion of the real storage on a zSeries system can be dedicated to a virtual machine. In this case, the storage of the virtual machine is real and the operating system running in the virtual machine can perform its own memory management without intervention by CP. Expanded storage can also be dedicated to a virtual machine.

CP also has facilities that allow the sharing of virtual pages by a number of virtual machines. A shared virtual page requires just one page of real storage, no matter how many virtual machines are sharing it, thereby economizing on real storage requirements. Domino doesn't use shared virtual pages.

### Minidisks

VM minidisks are virtual disk devices. They are implemented by partitioning a real volume into cylinder ranges that appear as separate disk volumes to the virtual machine. A minidisk can span a whole real disk volume. A real disk can also be dedicated to a virtual machine.

Minidisks can be shared or non-shared. If authorized, one virtual machine can link to a minidisk belonging to another virtual machine to access the data on it. Links can either read-only or read-write. When a minidisk is write-shared, some software is needed to manage access to the data. Domino data cannot be shared between Domino servers. However the Domino .tar file could be stored on a shared mindisk to reduce the amount of storage needed overall.

CP is able to cache the contents of minidisks in real or expanded storage to improve application response times.

### Temporary minidisks

Temporary minidisks are allocated from a defined pool of real disk storage, either when the virtual machine is logged on or by explicit definition. They last for the life of the virtual machine. When the virtual machine is logged off or the minidisk is detached, the temporary minidisk is destroyed.

### Virtual minidisks

Virtual minidisks (sometimes called *vdisks)* have similarities to temporary minidisks. But instead of being mapped to cylinders of real disk volumes, they are mapped into real storage by CP. This means that they avoid the need for disk I/O. The associated pages are managed by CP as part of its overall real memory management.

### Reader, punch, printer

These devices are not associated with real devices, but are implemented through the CP spooling subsystem. The spool files created or read on these devices can be transferred between virtual machines very easily.

### The console

The console is an important device for the virtual machine, as it is the primary user interface. When you log on to a virtual machine from a real 3270 terminal or 3270 emulator, the virtual console becomes associated with the real 3270. This allows you to enter CP commands and to IPL an operating system.

When an operating system is IPLed in a virtual machine, it will look for a device to be its system console. This device is often the one defined by the CONSOLE statement in the CP directory.

After the operating system is IPLed in a virtual machine, the console device is sometimes no longer required and it may be disconnected. The virtual machine can continue to function with a disconnected console.

### Channel-to-channel device

A virtual channel-to-channel device is implemented by CP entirely through software. Virtual I/O instructions are intercepted by CP and the data moved between memory buffers. This enables very high-speed communications between virtual machines.

### Virtual I/O

An operating system such as z/OS or VSE/ESA™ running in a virtual machine will issue normal S/390 I/O instructions to perform I/O. The operating system builds a string of Channel Command Words (CCWs) and issues a Start Subchannel (SSCH) instruction.

If the I/O is to a minidisk device, the virtual device number must be converted to a real device number and the virtual cylinder number must be converted to a real cylinder number. Also, the address of the data to be read or written must be converted to a real address. This process, called CCW Translation, is carried out transparently by CP.

By dedicating real storage and devices to a virtual machine, the processor-related overhead of CCW translation can be bypassed. A “preferred” guest can perform disk I/O without any intervention by CP.

### CMS

The Conversational Monitor System (CMS) is a unique zSeries operating system. It is IPLed in a virtual machine in the normal way, but is often a single user environment. You can think of it as the original time-sharing PC! CMS provides a rich application development and execution environment. It has powerful commands and tools that are useful to both developers and end users.

The REXX interpreter makes it easy to write command scripts and even whole applications. The XEDIT editor is a sophisticated editor with a large subcommand set. CMS Pipelines allows the concept of UNIX pipes.

Besides acting as a single-user system, server applications can also be written that run on CMS. Many z/VM products and tools are implemented in this way (the TCP/IP service machine and related daemon virtual machines, for example).

## 3.2 z/VM planning

This section describes some of the z/VM characteristics that you should be aware of.

- Base installation
- Memory
- Processors or CPUs
- Paging and spool space
- Managing the user directory
- Disk space for Linux
- Networking for z/VM and Linux—addressed in 5.2, “z/VM networking” on page 76

### 3.2.1 Base installation

Details on installing z/VM are beyond the scope of this redbook. For information about this topic, see *z/VM Guide for Automated Installation and Service, Version 4, Release 4.0*, GC24-6064. z/VM publications are on the Internet at:

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

Also, the z/VM product comes with a convenient one-page installation summary entitled *z/VM Installation and Service Summary.*

Normally z/VM 4.3 is installed on two 3390-3 DASDs, while z/VM 4.4 is installed on three.

### 3.2.2 Memory or storage

Traditionally, the term for memory used by zSeries personnel has been *storage*. Today, with the advent of Storage Area Networks, the term “storage” is used for disk space. z/VM commands still use the term “storage”. However, in this book, we will use the term “memory”.

The amount of memory that z/VM has is very important to the performance of almost any application. zSeries memory can be allocated as either central or expanded. Central memory is addressed at the byte level, while expanded memory is addressed a page at a time.

Expanded memory acts as a fast paging area. Central memory on non-zArchitecture machines (G5, G6, MP3000, and others) is limited to 2 GB due to 31-bit addressing.

To query central memory and expanded memory available to z/VM, the following commands are used.

```
q stor
STORAGE = 2097148K
q xstor
XSTORE= 2048M online= 2048M
XSTORE= 2048M userid= SYSTEM usage= 4% retained= 768M pending= 0M
XSTORE MDC min=0M, max=2048M, usage=4%
XSTORE= 1280M userid=  (none)  max. attach= 1280M
```

This shows that our z/VM system has approximately 2 GB of central storage and another 2 GB of expanded storage. A rule of thumb that has been suggested is to have a 3:1 ratio of central memory to expanded memory; however, this is not the case on our system.

### 3.2.3 Processors or CPUs

Traditionally the term for CPUs used by zSeries personnel has been *processors*. Also, the term *engines* is sometimes used.

Each zSeries box has multiple physical processors which come in a number of different flavors. The two flavors important to z/VM and Linux are central processor (sometimes called *standard engine*) and Integrated Facility for Linux (IFL). As they pertain to z/VM and Linux, IFLs perform identically to central processors on the same machine—the major difference is in price and licensing terms.

An LPAR either has CPUs dedicated to it, or shares CPUs with other LPARs. When the CPU is dedicated, the unused cycles are lost and cannot be used by the other LPARs. When you share CPUs, the LPAR is assigned a weight that determines what share of those CPUs can be used by the LPAR.

Virtual processors are assigned to virtual machines in a similar way. Adding virtual processors to a virtual machine allows the virtual machine to get more CPU resources. This is only useful when the operating system can take advantage of it.

It is not recommended to assign more virtual CPUs to a guest than you have real CPUs in your z/VM system. A minimum of two CPUs is recommended, dedicated or shared.

The number of CPUs can be queried from a privileged user ID with the following command:

```
q processors
PROCESSOR 00 MASTER
PROCESSOR 01 ALTERNATE
```

The maximum number of virtual CPUs that can be defined is specified in the user directory entry for each z/VM user ID. For example, the following line in the user directory entry specifies that a maximum of two CPUs can be defined to the z/VM user ID:

```
MACHINE ESA 2
```

The number of virtual CPUs can be set in one of two ways, either in the user directory, or interactively with the **DEFINE CPU** command. We show both ways here:

1. In the user directory:

   ```
   CPU 00 BASE
   CPU 01
   ```

2. Interactively, with the **DEFINE CPU** command. Here is an example of setting up a second virtual CPU:

   ```
   q cpus
   CPU 00  ID  FF02082370600000 (BASE)
   def cpu 01
   CPU 01 defined
   q cpus
   CPU 00  ID  FF02082370600000 (BASE)
   CPU 01  ID  FF02082370600000 STOPPED
   ```

   If you then IPL Linux, the new CPU will be used. For example, this is shown from z/VM using the **QUERY CPUS** command:

   ```
   CP Q CPUS
   CPU 00  ID  FF02082370600000 (BASE)
   CPU 01  ID  FF02082370600000
   ```

From Linux, the same processors can be queried through the /proc filesystem:

```
# cat /proc/cpuinfo
vendor_id       : IBM/S390
# processors    : 2
bogomips per cpu: 211.35
processor 0: version = FF,  identification = 020823,  machine = 7060
processor 1: version = FF,  identification = 020823,  machine = 7060
```

### 3.2.4 Page and spool space

z/VM requires adequate page space to perform well when more virtual memory is defined than physical memory. A rule of thumb, described in *Linux for S/390 and zSeries: ISP/ASP Solutions*, SG24-6299, is to have twice as much page space on DASD as the sum of your total virtual storage (which includes z/VM and all Linux guests) and virtual disk.

For example, if your z/VM has 2 GB of central storage and you plan to have 12 Linux guests each with 512 MB, then you should have 16 GB of paging space (`2*(2+12*0.5)`). That would equate to 7 3390-3 paging volumes (or *packs*).

Another rule of thumb says that z/VM should only use an average of 30% or less of its page space. If z/VM runs out of page space, it will abend with a code of PGT004.

The amount of page space currently allocated and being used can be queried with the **QUERY ALLOC PAGE** command:

```
q alloc page
EXTENT EXTENT  TOTAL  PAGES   HIGH    %
VOLID  RDEV  START    END  PAGES IN USE   PAGE USED
------ ---- ------ ------ ------ ------ ------ ----
VMPAG1 0202      1   3338 600840    526    556   1%
VMPAG2 0203      1   3338 600840      0      0   0%
                           ------ ------         ----
SUMMARY                     1202K    526           1%
USABLE                      1202K    526           1%
```

This shows that this z/VM system is using almost no page space. However, only a single Domino server is running. If your installation will have several Domino servers you will need to add page space. To add additional page, spool and temporary disk space, see 3.3.6, “Add page, spool and temporary disk space” on page 44.

### 3.2.5 Disk space for z/VM and Linux

Traditionally, the term for disk space used by zSeries personnel has been DASD. A popular format is that of 3390-3s, which are about 2.3 GB when formatted for Linux. Often these are referred to as “mod-3s” or simply “packs”. z/VM is installed on 2 or 3 3390-3s plus what is needed for page and spool space. Each Linux image with Domino for Linux on zSeries code is often installed on two 3390-3s plus whatever space is needed for the Notesdata directory. For details, see Chapter 4, “Disk configuration” on page 53.

For example, we needed at least 66 packs (3390-3 DASD) for this project: six for z/VM and 60 for the Linux guests. z/VM 4.3 requires two install packs and we dedicated two packs for page space and one for spool/tdisk. Also, part of one pack was used for the Linux user ID 191 disks for some CMS read/write space. Each of the three Linux user IDs was given 20 packs: two to install Linux and Domino, and 18 for a large volume group for Notesdata and translogs. We could have used other DASD sizes including 3390-9 and 3390-27.

In retrospect we would have used one volume for notesdata, one volume for translogs, and 16 volumes for a large volume group for the mail directories. You may also want to reserve a volume for diagnostic data such as Domino NSDs and logs. By default these are placed in the IBM_TECHNICAL_SUPPORT directory within notesdata, but you might want to isolate them for manageability.

zSeries also has the ability to utilize Storage Area Networks (SAN) with the new FCP support. This is a relatively new technology, not widely utilized in production. There is an introductory Redpaper dealing with this topic entitled *Getting Started with zSeries Fibre Channel Protocol*, REDP-0205. It is available on the Web at:

http://www.redbooks.ibm.com/redpapers/pdfs/redp0205.pdf

### 3.2.6 Defining user IDs for Linux images

The CP or user directory is commonly maintained in one of two ways:

- A separately-priced product, such as IBM DirMaint or CA VM:Direct
- Manually editing the user directory file

When CMS under VM was a popular end-user operating system, hundreds or thousands of users and minidisks often had to be maintained. Directory maintenance products such as DirMaint or VM:Direct were popular due to the complexity of managing this number of directory entries.

However, with Linux on zSeries, the number of directory entries is often in the tens or even less than ten. When this is the case, manually editing the USER DIRECT file and applying the changes by the DIRECTXA command is simple and the complexity is not significant. Thus we recommend manually editing the USER DIRECT file unless a directory maintenance product is already in place and the necessary skills are available.

## 3.3 z/VM tasks

Some of the common z/VM system administration tasks are described in this section:

- Get help
- Determine who is on the system
- Determine how busy the system is
- Determine what DASD you have
- Format and add DASD to the system
- Add page, spool and temporary disk space
- Add Linux user IDs
- Update the SYSTEM CONFIG file
- Modify the logon screen

A good tutorial on z/VM basics is provided in Chapter 1 of *Linux for zSeries and S/390: Large Scale Linux Deployment*, SG24-6824, on the ITSO Web site at:

http://www.redbooks.ibm.com/abstracts/sg246824.html

In the following sections, we provide only high level descriptions of these tasks. To help fill in the many gaps, refer to *CP Planning and Administration*, SC24-6043 and *CP Command and Utility Reference*, SC24-6008.

For networking issues and tasks, see 5.2, “z/VM networking” on page 76.

### 3.3.1 Get help

The z/VM help file is thorough, although traversing it is not always easy. Since CP (but not CMS) commands are available to Linux images that are running, a useful command is **HELP CP MENU**:

```
help cp menu
...
To display a Help panel, move the cursor to any character
of the name and press the ENTER key or the PF1 key.
An asterisk (*) preceding the name indicates a MENU panel.
A colon (:) preceding the name indicates a TASK panel.

*CPQUERY  *XLINK    CPACcess  DISCARD   LOCate    RESTART   SYStem
*CPSET    *XSPOOL   CPCAche   DISConn   LOCATEVM  RETAIN    TAg
```

```
*CPUTIL   :DUMPS    CPHX      DISPlay   LOCK      REWind    TERMinal
*DEFINE   :DYNIO    CPLISTfi  DRain     LOGoff    SAVESEG   TRace
*DELETE   :HELP     CPRELeas  DUmp      Logon     SAVESYS   TRANsfer
*DETACH   #CP       CPTRAP    DUPlex    Message   SCREen    TRSAVE
*DISABLE  ACNT      CPTYPE    ECho      MODify    SEND      TRSOurce
*DISPLAY  ACTivate  CPU       ENable    MONitor   SET       UNCOUPLE
*DUMP     ASSOCiate CPVLoad   EXTernal  MSGNOH    SHUTDOWN  UNDEDicat
*ENABLE   ASTERISK  CPXLoad   FLASHCopy NOTReady  SIGnal    UNDIAL
*INDICATE ATTach    CPXUnload Flush     ORDer     SILENTly  UNLOCK
*MODIFY   ATTN      DEActive  FORCE     PER       SLeep     UTILITY
*MONITOR  AUTOLOg   DEDicate  FOrward   POUNDCP   SMsg      VARY
*PURGE    BAckspac  DEFine    FRee      PURge     SNAPDUMP  VDelete
*RDEVICE  Begin     DEFSEG    GIVE      Query     SPAce     VInput
*SPTAPE   CHange    DEFSYS    HALT      READY     SPool     VMDump
```

From this screen, you can navigate your cursor with the arrow or tab keys to get help on each command. Also note that the commands listed on the left side of the screen with a leading asterisk each have a menu of their own. Because **CP QUERY** and **CP SET** commands have many possible parameters, those help menus are quite useful.

In addition to the help system, the z/VM manuals provide extensive information and are available online:

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

### 3.3.2 Determine who is on the system

You can determine who is on a z/VM system with the CP **query names** command. This is analogous to the Linux **who** command.

For example:

```
q n
NPM      - DSC , ESAMAP   -L0003, ESAWEB01 - DSC , LINUXC   - DSC
LINUXA   - DSC , ESAADMIN - DSC , ESASERVE - DSC , AUTOUNO  - DSC
VMAINT   - DSC , LFSSERV  - DSC , LOGNSYSC - SYSC, NETVIEW  - DSC
SNMPD    - DSC , AUTONPM  - DSC , CFCC2    - DSC , CFCC1    - DSC
TCPIP    - DSC , RSCS     - DSC , SFSTEST  - DSC , VMSERVU  - DSC
VMSERVS  - DSC , VMSERVR  - DSC , VMID     - DSC , VTAM     - DSC
YVETTE   - DSC , VMSERVER - DSC , TOOLS    - DSC , DATAMOVE - DSC
DIRMAINT - DSC , TODEVENT - DSC , EYE      - DSC , ISPVM    - DSC
PVM1     - DSC , PVM      - DSC , SMART    - DSC , GCS      - DSC
RACFVM   - DSC , OPERSYMP - DSC , DISKACNT - DSC , EREP     - DSC
OPERATOR - DSC , RACFRSM  - DSC , ESATCP   - DSC , MAINT    -L000A
ESAWRITE - DSC , LINUXB   -L0008
VSM     - TCPIP
VSM     - VTAM
```

The output shows which users are logged on and what their virtual terminal addresses are. When the user IDs are disconnected, it shows DSC.

### 3.3.3 Determine how busy the system is

At a high level, the **INDICATE** command allows you to monitor what is happening your system. For example:

```
ind
AVGPROC-009% 02
XSTORE-000001/SEC MIGRATE-0000/SEC
MDC READS-000039/SEC WRITES-000006/SEC HIT RATIO-061%
```

```
STORAGE-028% PAGING-0001/SEC STEAL-000%
Q0-00000(00000)                          DORMANT-00043
Q1-00000(00000)           E1-00000(00000)
Q2-00000(00000) EXPAN-001 E2-00000(00000)
Q3-00006(00000) EXPAN-001 E3-00000(00000)

PROC 0000-008%          PROC 0001-010%

LIMITED-00000
```

See Chapter 12, “Performance monitoring and tuning” on page 315, for more details on determining how busy the system is.

### 3.3.4 Determine what DASD and minidisks you have

You can query the DASD that your virtual machine has with CP’s **QUERY DASD** command. From an emulated 3270 session with Linux running, you can prefix the command with **#CP**. For example:

```
#cp q da
DASD 0190 3390 VMLRES R/O        107 CYL ON DASD  3708 SUBCHANNEL = 000A
DASD 0191 3390 VMLU1A R/W         50 CYL ON DASD  37C9 SUBCHANNEL = 000E
DASD 019D 3390 VMLP1P R/O        102 CYL ON DASD  3748 SUBCHANNEL = 000D
DASD 019E 3390 VMLP1P R/O        300 CYL ON DASD  3748 SUBCHANNEL = 000B
DASD 019F 3390 VMLP1P R/O        400 CYL ON DASD  3748 SUBCHANNEL = 000C
DASD 0200 3390 LX1518 R/W       3338 CYL ON DASD  1518 SUBCHANNEL = 000F
DASD 0201 3390 LX1558 R/W       3338 CYL ON DASD  1558 SUBCHANNEL = 0010
...
DASD 0213 3390 LX15DD R/W       3338 CYL ON DASD  15DD SUBCHANNEL = 0022
DASD 0300 9336 (VDSK) R/W     100000 BLK ON DASD  VDSK SUBCHANNEL = 0024
...
DASD 0309 9336 (VDSK) R/W     100000 BLK ON DASD  VDSK SUBCHANNEL = 002D
DASD 0592 3390 VMLNET R/O         67 CYL ON DASD  3709 SUBCHANNEL = 0023
```

This command is useful from a Linux ID to see what minidisks you have as read/write. Linux user IDs commonly have only class G privilege. Because the user does not have class B privilege, the previous command is really a **QUERY VIRTUAL DASD** which displays the status of all DASDs accessible to your virtual machine, including temporary disks (T-disks) and virtual disks in storage.

For users that do have class B privilege, such as MAINT, other **QUERY DASD** commands are useful, such as:

**QUERY DASD FREE** Shows all free DASD that are not currently in use by a user or the system and is specified as neither active nor offline

**QUERY DASD ALL** Shows also offline and free DASD

The command **QUERY DISK** will show you what CMS disks you have accessed, but that is only available when CMS is running, not Linux.

### 3.3.5 Format and add DASD to the system

DASD must first be added to the system and formatted before it can be used by Linux user IDs. You can query what DASD is available for use by the system with the **QUERY DASD FREE** command.

The following example checks for free DASD through **QUERY DASD FREE** command, attaches the DASD to the system with the **ATTACH** command, and then formats the DASD using the **CPFMTXA** command.

Normally, you can just format cylinder 0 for z/VM's use, and use the Linux **dasdfmt** command to format the majority of the disk for Linux:

```
q da free
DASD 1516 LX1516  , DASD 1517 LX1517  , DASD 1556 LX1556  , DASD 1557 LX1557
DASD 1596 LX1596  , DASD 1597 LX1597  , DASD 15C6 LX15C6  , DASD 15C7 LX15C7
att 1516 *
DASD 1516 ATTACHED TO MAINT 1516 WITH DEVCTL
cpfmtxa
ENTER FORMAT, ALLOCATE, LABEL, OR QUIT:
format
ENTER THE VDEV TO BE PROCESSED OR QUIT:
1516
ENTER THE CYLINDER RANGE TO BE FORMATTED ON DISK 0204 OR QUIT:
0 0
ENTER THE VOLUME LABEL FOR DISK 0204:
lx1516
CPFMTXA:
FORMAT WILL ERASE CYLINDERS 00000-0000 ON DISK 0204
DO YOU WANT TO CONTINUE? (YES | NO)
yes
...
ICK00001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
          10:39:02    09/09/03

ENTER INPUT COMMAND:
 END

ICK00002I ICKDSF PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0
ENTER ALLOCATION DATA
TYPE CYLINDERS
................
end
```

Add the DASD by volume serial name (VOLSER) to the user volume list in the SYSTEM CONFIG file, so it will still be attached to the system after the next system IPL. See 3.3.8, "Update the SYSTEM CONFIG file" on page 49 for details on how to update this file. Here is an example of adding a new volume.

```
/*********************************************************************/
/*                        User_Volume_List                           */
/*********************************************************************/

    User_Volume_List VMLRAC
    User_Volume_List VMLRAB
    User_Volume_List VMLU1R
    User_Volume_List VMLU2R
    User_Volume_List VMLU1A
    User_Volume_List VMLP1P
    User_Volume_List VMLNET
    User_Volume_List LX1512
    User_Volume_List LX1516
    User_Volume_List LX1518
    ...
```

The new DASD should now be available for use. The volume name (VOLSER) can be used in the USER DIRECT file or by a directory maintenance product.

## 3.3.6 Add page, spool and temporary disk space

In the example that follows, two volumes at addresses 202 and 203 are dedicated to page space and another volume at address 204 is dedicated to spool and temporary disk space. On a real z/VM system with many Linux images, multiple paging volumes will be needed (see 3.2.4, “Page and spool space” on page 38).

The steps involved in this task are as follow:

- Attach, format and allocate the DASD
- Make the DASD CP-owned in the SYSTEM CONFIG file
- Re-IPL the system to verify the new setup
- Remove page, spool and tdisk space from res pack

### Attach, format, and allocate the DASD

This is normally done from the MAINT user ID. The DASD must first be attached to MAINT, not to the SYSTEM:

```
q 202-204
DASD 0202 ZVM202 , DASD 0203 ZVM203 , DASD 0204 ZVM204
att 202 *
DASD 0202 ATTACHED TO MAINT 0202 WITH DEVCTL
att 203 *
DASD 0203 ATTACHED TO MAINT 0203 WITH DEVCTL
att 204 *
DASD 0204 ATTACHED TO MAINT 0204 WITH DEVCTL
```

The DASD must first be formatted with the **CPFMTXA** command (CP can abend if it tries to use unformatted PAGE, SPOL or DRCT space). The allocation of the cylinders can then be set. The first volume is allocated paging space:

```
cpfmtxa
ENTER FORMAT, ALLOCATE, LABEL, OR QUIT:
format
ENTER THE VDEV TO BE PROCESSED OR QUIT:
202
ENTER THE CYLINDER RANGE TO BE FORMATTED ON DISK 0202 OR QUIT:
0 end
ENTER THE VOLUME LABEL FOR DISK 0202:
vmpag1
CPFMTXA:
FORMAT WILL ERASE CYLINDERS 00000-03338 ON DISK 0202
DO YOU WANT TO CONTINUE? (YES | NO)
yes
HCPCCF6209I INVOKING ICKDSF.
...
CYLINDER ALLOCATION CURRENTLY IS AS FOLLOWS:
          TYPE     START     END     TOTAL
          ----     -----     ---     -----
          PERM     0         3337    3338
...
ENTER ALLOCATION DATA
TYPE CYLINDERS
.................
page 1 3338
end
...
CYLINDER ALLOCATION CURRENTLY IS AS FOLLOWS:
TYPE     START     END     TOTAL
----     -----     ---     -----
```

```
PERM     0         0       1
PAGE     1         3338    3338
```

The same process is done for the DASD at address 203, but the label is VMPAG2. A similar process is done for the DASD at address 204; however, 2200 cylinders are allocated for spool (SPOL) and the remainder for temporary disk (TDSK):

```
cpfmtxa
ENTER FORMAT, ALLOCATE, LABEL, OR QUIT:
format
ENTER THE VDEV TO BE PROCESSED OR QUIT:
204
ENTER THE CYLINDER RANGE TO BE FORMATTED ON DISK 0204 OR QUIT:
0 end
ENTER THE VOLUME LABEL FOR DISK 0204:
vmspl1
...
ENTER ALLOCATION DATA
TYPE CYLINDERS
.................
spol 1 2200
tdsk 2201 3338
end
...
CYLINDER ALLOCATION CURRENTLY IS AS FOLLOWS:
          TYPE     START     END     TOTAL
          ----     -----     ---     -----
          PERM     0         0       1
          SPOL     1         2200    2200
          TDSK     2201      3338    1138
...
```

Then issue the **QUERY ALLOC** command:

```
q alloc
DASD 0200 430RES 3390 CKD-ECKD (UNITS IN CYLINDERS)
     TDISK TOTAL=000184 INUSE=000000 AVAIL=000184
     PAGE  TOTAL=000000 INUSE=000000 AVAIL=000000
     SPOOL TOTAL=000178 INUSE=000131 AVAIL=000047
     DRCT  TOTAL=000020 INUSE=000001 AVAIL=000019, ACTIVE
DASD 0201 430W01 3390 CKD-ECKD (UNITS IN CYLINDERS)
     TDISK TOTAL=000000 INUSE=000000 AVAIL=000000
     PAGE  TOTAL=000000 INUSE=000000 AVAIL=000000
     SPOOL TOTAL=000000 INUSE=000000 AVAIL=000000
     DRCT  TOTAL=000000 INUSE=000000 AVAIL=000000
DASD 0202 VMPAG1 3390 CKD-ECKD (UNITS IN CYLINDERS)
     TDISK TOTAL=000000 INUSE=000000 AVAIL=000000
     PAGE  TOTAL=003338 INUSE=000004 AVAIL=003334
     SPOOL TOTAL=000000 INUSE=000000 AVAIL=000000
     DRCT  TOTAL=000000 INUSE=000000 AVAIL=000000
DASD 0203 VMPAG2 3390 CKD-ECKD (UNITS IN CYLINDERS)
     TDISK TOTAL=000000 INUSE=000000 AVAIL=000000
     PAGE  TOTAL=003338 INUSE=000000 AVAIL=003338
     SPOOL TOTAL=000000 INUSE=000000 AVAIL=000000
     DRCT  TOTAL=000000 INUSE=000000 AVAIL=000000
DASD 0204 VMSPOL 3390 CKD-ECKD (UNITS IN CYLINDERS)
     TDISK TOTAL=001138 INUSE=000000 AVAIL=001138
     PAGE  TOTAL=000000 INUSE=000000 AVAIL=000000
     SPOOL TOTAL=002200 INUSE=000025 AVAIL=002175
     DRCT  TOTAL=000000 INUSE=000000 AVAIL=000000
IPL NUCLEUS ACTIVE ON VOLUME 430RES
```

The DASD can be added to the SYSTEM CONFIG file, in which case a re-IPL is necessary—or it can be added dynamically

### Adding the DASD to the SYSTEM CONFIG file

The new volumes for page and spool space will be owned by CP and are identified by their volume labels, so modify the SYSTEM CONFIG file (see 3.3.8, “Update the SYSTEM CONFIG file” on page 49) using XEDIT and modify the following lines:

```
CP_Owned   Slot   1  430RES
CP_Owned   Slot   2  430W01
CP_Owned   Slot   3  VMPAG1
CP_Owned   Slot   4  VMPAG2
CP_Owned   Slot   5  VMSPOL
CP_Owned   Slot   6  RESERVED
CP_Owned   Slot   7  RESERVED
```

It is good to leave a least two slots defined as RESERVED for reasons described in the next section.

### Adding the DASD dynamically

If the CP_OWNED Slots were already allocated in the SYSTEM CONFIG file as Reserved, then the devices could be added dynamically. For example:

```
q cpowned
Slot  Vol-ID  Rdev  Type   Status
   1  VM1RES  1434  Own    Online and attached
   2  VM1PST  1474  Own    Online and attached
   3  ------  ----  -----  Reserved
   4  ------  ----  -----  Reserved
```

Slots 3 and 4 have been reserved. To utilize slot 3, the following commands can be used:

```
q dasd vmpag1
dasd 1423 VMPAG1
define cpowned slot 3 vmpag1
att 1423 system
q cpowned
Slot  Vol-ID  Rdev  Type   Status
   1  VM1RES  1434  Own    Online and attached
   2  VM1PST  1474  Own    Online and attached
   3  VMPAG1  1423  Own    Online and attached
   4  ------  ----  -----  Reserved
```

Now slot 3 is utilized. You can update the SYSTEM CONFIG file with this change. After the next IPL, the new page pack will still be used.

Note that for CP-owned volumes that have spool on them, that you *cannot* change the slot order of those volumes once they are in use. For example, consider the following:

```
CP_Owned    Slot    1   430RES
CP_Owned    Slot    2   430W01
CP_Owned    Slot    3   VMPAG1
CP_Owned    Slot    4   VMSPOL
CP_Owned    Slot    5   Reserved
```

If you want to you remove the page pack VMPAG1, you must leave the spool pack, VMSPOL, in slot 4.

## Re-IPL the system to verify the new setup

Now that the new page and spool space is established, the system must be re-IPLed for it to take effect. Be sure that all Linux images are shut down and logged off.

```
shutdown reipl
```

The system should shut down and come back within a minute or two. Log back on to MAINT.

## Remove page, spool and tdisk space from RES pack

For better performance, it has been recommended to remove page, spool, and temporary disk space from the residence pack. You can choose which of these spaces to remove from the RES pack. TDSK space is the easiest to remove, but perhaps the least critical.

Page space can just be removed if you don't mind re-IPLing again (if you do not re-IPL, the page space must be drained). It is more complex to remove spool space; the saved segments must be rebuilt. Therefore, it is recommended that only PAGE and TDSK space are removed followed by a reboot. First determine which cylinders on the RES pack are being used with the **QUERY ALLOC MAP** command:

```
q alloc map
            EXTENT EXTENT                        % ALLOCATION
VOLID  RDEV  START    END  TOTAL IN USE   HIGH USED TYPE
------ ---- ------ ------ ------ ------ ------ ---- -------------
PAKA01 C510      1      4      4      1      1  25% DRCT ACTIVE
               100   1099 180000    536    566   1% PAGE
              1100   2099 180000   7117  10893   3% SPOOL
              2100   2299    200      0      0   0% TDISK
ZMC511 C511      1   3338 600840      0      0   0% PAGE
ZMC512 C512      1   3338 600840      0      0   0% PAGE
ZMC513 C513      1   3338 600840   3308   3308   1% SPOOL
ZMC514 C514      1   3338   3338      0      0   0% TDISK
```

Therefore, cylinders 100-1099 and 2100-2299 should be assigned as PERM space.

**Important:** Be very careful when changing the assignment of the z/VM RESpack. To be really safe, shut down the system and back up the RESpack before the next step.

Following are the subcommands to complete the reallocation:

```
CPFMTXA
ALLOC
123
PAKA01
perm 100 1099
perm 2100 2299
end
```

The system is shut down using the command **SHUTDOWN REIPL**. You will lose your 3270 session unless you are at the console. When the system comes back, the **QUERY ALLOC MAP** command is used. Notice that the RES pack has SPOOL space, but no longer has TDSK or PAGE space on it. Since PAGE and TDSK are reused after each system IPL, this process works. SPOOL space is retained across IPLs and cannot be modified using this process.

```
shutdown reipl
...
q alloc map
            EXTENT EXTENT                        % ALLOCATION
VOLID  RDEV  START    END  TOTAL IN USE   HIGH USED TYPE
------ ---- ------ ------ ------ ------ ------ ---- -------------
```

```
PAKA01 C510      1      4      4      1      1  25% DRCT ACTIVE
               1100   2099 180000   7127  10893   3% SPOOL
ZMC511 C511      1   3338 600840    536    566   1% PAGE
ZMC512 C512      1   3338 600840      0      0   0% PAGE
ZMC513 C513      1   3338 600840   3355   3355   1% SPOOL
ZMC514 C514      1   3338   3338   1158   3338  34% TDISK
```

## 3.3.7 Add Linux user IDs

As discussed, there are two different approaches to maintaining the user directory: with a directory maintenance product, or by editing the USER DIRECT file. The example shown in "The CP or user directory" on page 31 is a DirMaint <user ID> DIRECT file taken from our system. DirMaint is discussed briefly in the next section.

### Defining user IDs through DirMaint

Following are some of the more straightforward subcommands:

- **DIRM ADD <user ID>** - to add a user
- **DIRM FOR <user ID> GET** to get a directory entry into MAINT's reader
- **DIRM FOR <user ID> REP** - to replace a modified directory entry
- **DIRM FOR <user ID> REV** - to get a read-only directory entry into MAINT's reader
- **DIRM DIRMAP** - to get a report detailing the current DASD utilization on the system.

DirMaint can be used in more sophisticated ways. DirMaint prototype entries (PROTODIR files) allows users to be created similar to others. The DirMaint LIKE option is useful when allocating disks. You can give Linux user IDs disks that are already formatted, based on copying from a single formatted disk. This can be helpful for cloning Linux images, or when you want to have swapping on real disks.

For more details on DirMaint, see *Linux for IBM @server zSeries: ISP/ASP Solutions*, SG24-6299.

### Defining user IDs in the USER DIRECT file

The most straightforward method of maintaining the z/VM user directory is to use the USER DIRECT file and the **DIRECTXA** command. This file is on the MAINT 2CC disk, which is accessed as minidisk C by default. When you edit the USER DIRECT file, you are changing the *source* of the user directory. It is the **DIRECTXA** command that reads the source and writes to the directory space on the RES pack. An important statement in the file is the DIRECTORY control statement:

```
DIRECTORY 123 3390 430RES
```

This statement tells **DIRECTXA** to write the directory to the virtual 123 disk, which has a label of 430RES.

Remember when we looked at the system's allocation map? The first entry is the actual directory on disk:

```
q alloc map
            EXTENT EXTENT                        % ALLOCATION
VOLID  RDEV  START    END  TOTAL IN USE   HIGH USED TYPE
------ ---- ------ ------ ------ ------ ------ ---- -------------
PAKA01 C510      1      4      4      1      1  25% DRCT ACTIVE
```

The **DIRECTXA** command first performs syntax checking on the USER DIRECT file. If it finds something wrong, it will report it. If everything is correct, you will see the following message, which means the changes were brought online.

```
directxa user
z/VM USER DIRECTORY CREATION PROGRAM - VERSION 4 RELEASE 3.0
EOJ DIRECTORY UPDATED AND ON LINE
```

Regarding the contents of the USER DIRECT file: the default values in the file installed with the system are all fine, and an example of a directory entry is described in “The CP or user directory” on page 31.

An additional aspect of the USER DIRECT file worth mentioning is PROFILEs. A *profile* is a template that can be used over again. When you want to use the profile, you use an INCLUDE statement. There is a default profile named IBMDFLT. You may want to adapt a new profile for Linux user IDs.

All aspects of user IDs that are to be common to Linux can be kept in this profile. An example of a Linux profile that defines several types of networking interfaces is as follows:

```
PROFILE LINDFLT
  IPL CMS
  MACH ESA 2
  IUCV ANY
  IUCV ALLOW
  CPU 00 BASE
  CPU 01
  CRYPTO APVIRT
  SPOOL 000C 2540 READER *
  SPOOL 000D 2540 PUNCH A
  SPOOL 000E 1403 A
  CONSOLE 009 3215 T
  SPECIAL 150 CTCA
  SPECIAL 151 CTCA
  SPECIAL 500 HIPER 3 SYSTEM GUESTLAN
  LINK TCPMAINT 592 592 RR
  LINK MAINT 0190 0190 RR
  LINK MAINT 019D 019D RR
  LINK MAINT 019E 019E RR
  LINK MAINT 0402 0402 RR
  LINK MAINT 0401 0401 RR
  LINK MAINT 0405 0405 RR
```

Then, when you want to define a Linux user ID, you include the profile in the user directory entry. For example:

```
USER LINUXB XXXXXXXX 512M 1G       G
 INCLUDE LINDFLT
 MDISK 200 3390 0001 3338 LX1516 MR RPASS WPASS
 MDISK 201 3390 0001 3338 LX1556 MR RPASS WPASS
```

PROFILEs are a great tool to use for keeping Linux user IDs standard.

### 3.3.8 Update the SYSTEM CONFIG file

There are three CP DISKs by default. Typically only the MAINT CF1 disk is used, but the other two can be used for backup.

Now link the MAINT CF1 disk read-write. To do this, use the **cprel** command to release it.

```
q cpdisk
```

```
Label  Userid   Vdev Mode Stat Vol-ID Rdev Type   StartLoc     EndLoc
MNTCF1 MAINT    OCF1  A   R/O  430RES 0200 CKD         391        435
MNTCF2 MAINT    OCF2  B   R/O  430RES 0200 CKD         436        480
MNTCF3 MAINT    OCF3  C   R/O  430RES 0200 CKD         481        525
cprel a
CPRELEASE request for disk A scheduled.
HCPZAC6730I CPRELEASE request for disk A completed.
```

Now the disk can be linked and accessed read-write.

```
link * cf1 cf1 mr
acc cf1 f
```

There are a few items you will probably want to update immediately. The System Identifier is the name of the z/VM system that shows up in the lower right portion of every 3270 session.

```
/*********************************************************************/
/*                   System_Identifier Information                   */
/*********************************************************************/

    System_Identifier_Default ZVM6
```

The default size of the retrieve stack (similar to the Linux **history**) is only seven commands. It is recommended that you make this value larger.

For example:

```
Features ,
   Disable ,                        /* Disable the following features */
     Set_Privclass ,                /* Disallow SET PRIVCLASS command */
     LogMsg_From_File ,             /* No LOGMSG from SYSTEM LOGMSG   */
     Auto_Warm_IPL ,                /* Prompt at IPL always           */
     Clear_TDisk   ,                /* Don't clear TDisks at IPL time */
   Retrieve ,                       /* Retrieve options               */
     Default  99 ,                   /* Default.... default is 7       */
     Maximum  99 ,                   /* Maximum.... default is 7       */
...
```

If you know the address of the console, update the console definitions. For example, the following updates the console to address 0009:

```
/*********************************************************************/
/*                     Console Definitions                           */
/*********************************************************************/

 Operator_Consoles            0009 0021 0022 0023 0E20 0E21 1020
 Emergency_Message_Consoles   0009 0021 0022 0023 0E20 0E21 1020
```

Save the file and verify the integrity of the changes via the **CPSYNTAX** command which is on the MAINT 193 disk:

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

Now release and detach the CF1 minidisk and assign it back to the system:

```
rel f
det cf1
cpacc * cf1 a
CPACCESS request for mode A scheduled.
Ready; T=0.01/0.01 08:56:36
HCPZAC6732I CPACCESS request for MAINT's OCF1 in mode A completed.
```

### 3.3.9 Modify the logon screen

The default z/VM logon screen can be customized. To do so, access the system configuration disk as read/write (see 3.3.8, “Update the SYSTEM CONFIG file” on page 49) and edit the file LOCAL LOGO. After you modify the file, **CPACCESS** the configuration minidisk and re-IPL the system. The logon panel will show the contents of the modified file.

4

# Disk configuration

In this chapter, we discuss aspects of DASD configuration that must be considered when running Domino under Linux in the zSeries environment. We discuss concepts and factors which are of concern at the z/VM and Linux levels. We also describe how to organize the Domino data to get the most out of your Domino for Linux on zSeries infrastructure by exploiting z/VM.

## 4.1 Introduction

DASD planning is a very important task and has a great influence on the overall performance of the Domino server. During the development of this redbook, we had a great deal of discussion about how to configure and lay out the file systems. Linux can be run under z/VM or as a standalone operating system in LPAR mode. In this project, we worked primarily with Linux under VM. However, most of our discussions about disk issues apply to both VM and standalone Linux, and relate to zSeries hardware configuration. z/VM, with its minidisk architecture and Linux's logical volume manager (LVM) capability, offers many choices of how to configure your DASD.

In the following sections, we discuss the z/VM and Linux perspective of DASD management. We introduce the concept of VM minidisks and the use of virtual disks for temporary data. Furthermore, we highlight how to configure the LVM under Linux, so that you can group multiple DASD volumes into one logical volume for Linux and the Domino server.

## 4.2 Mount points

In Linux and UNIX systems, *mount points* are the locations where physical hard drives are attached. The entire directory tree structure spans one or more mount points, depending on the size of the devices used in on the system. A directory or sub-directory may not span multiple mount points.

Different mount points can have different filesystem types, such as journaled or non-journaled. Since these mount points are the attachment points for hard drives (DASD) or logical volumes (multiple physical drives as a single logical drive), the drives may have different physical characteristics.

Proper planning of directory layouts through mount points allows you to take advantage of better performing hardware and filesystems.

### 4.2.1 Shared filesystems

VM allows you to share DASD across guest virtual machines (typically in read-only mode). A VM DASD volume often equates to a filesystem. Table 4-1 lists some directories you might want to share among your various Linux guest machines.

*Table 4-1 Filesystems to share*

| Linux directory | How used |
|---|---|
| /bin | Linux shells and commands |
| /boot | Linux Kernel and kernel support items |
| /lib | Shared libraries that are dynamically linked |
| /opt | Product package code for additional software products typically each in their own subdirectories |
| /sbin | Programs requiring root or superuser access |
| /sbin.initd | Scripts for Linux programs that are started and stopped on Linux IPL and shutdown, respectively. |
| /usr/bin | User applications |

The Domino .tar file which contains the Domino install code can be shared between Linux guests, but the installed executables in /opt/lotus cannot easily be shared. During the installation some files unique to the Domino server being installed are stored in /opt/lotus.

Keep in mind that we are sharing at the minidisk level. Directories that span multiple minidisks grouped into logical volumes are not candidates for sharing.

**Note:** You cannot share directories with Domino databases inside, because a Domino server needs exclusive access to its databases.

### 4.2.2 Minidisks

The purpose of a VM system is to make available to users of VM a virtual computer which appears to users as though they have full, exclusive use of all the resources of the machine. The resources that a user has access to are defined in the directory entry for the user.

One of the resources that can be configured is DASD. Although there are a large range of 3390 DASD types, the most common is the 3390 Model 3 DASD pack, which has about 2.3 GB of space. Less common but also frequently used is the 3390 Model 9 DASD, with 6.9 gigabytes of formatted space. A user may not need all of this space but only a portion of it.

VM employs the concept of minidisks, whereby physical DASD can be split into smaller sizes, or minidisks, and assigned to individual VM user IDs or guest systems. Minidisks can be as small as one cylinder or as large as all the cylinders on a DASD device. One important thing to keep in mind is that the size of a minidisk cannot exceed the number of cylinders on a disk. Cylinder 0 is used exclusively by the operating system. Minidisks cannot span physical DASD devices.

A *minidisk* is a contiguous portion of a physical DASD, similar to a partition in the Windows world that is defined for exclusive use by a specific owner user ID. When the minidisk is defined, it is given an address in the same way that a DASD pack is defined with an address. To users, it appears as though they have access to a physical disk of a specific size. What users do with this disk is based on what their needs are. To users, this is just another DASD device over which they have full control.

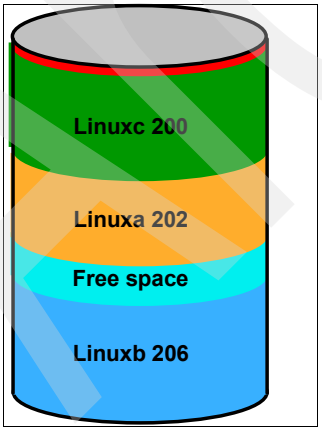



*Figure 4-1 Multiple minidisks on one DASD*

When we run Linux under VM, we are in essence running Linux natively on a virtual mainframe with virtual resources such as DASD, memory, and CPUs. From the perspective of Linux, it appears that these resources are real.

With regard to minidisks, Linux views them as though they are hard drives that can be addressed natively. When we install and configure Linux, the drives are formatted and controlled as they would be on any other Linux system, complete with boot blocks and so on.

### Minidisk size

The most common DASD type for a IBM mainframe is the 3390. Actual 3390 DASD has not been manufactured for many years. It may be possible that this type of DASD is still in use, but it is far more likely that the DASD in use today is emulating a 3390. 3390 DASD is commonly emulated by hardware such as RVA and ESS.

When looking at the model number for a 3390 device, the number is roughly equivalent to the amount of data in gigabytes that the device can store. 3390 mod 3 (2.3 GB) and mod 9 (6.9 GB) are most commonly used on z/OS. When running Linux on z/VM, other sizes can be used. A new, larger model of DASD can be emulated, although no physical hardware of this type was ever made; this is the 3390-27 or mod-27 device.

The models listed in Table 4-2 can be emulated and used by Linux.

*Table 4-2 DASD model and capacity*

| Model | Number of cylinders | Approximate capacity native | Approximate capacity after Linux dasdfmt |
|---|---|---|---|
| 3390-1 | 1113 | 924 MB | 789 MB |
| 3390-2 | 2226 | 1.8 GB | 1.54 GB |
| 3390-3 | 3339 | 2.7 GB | 2.3 GB |
| 3390-9 | 10017 | 8.12 GB | 6.9 GB |
| 3390-27 | 32000 | 25.9 GB | 22.1 GB |

In this guide we take the approach of dedicating whole DASD volumes for Linux systems. However, these devices are split up into minidisks owned by individual Linux guests. If your DASD has been “gen’d shared” in your IOCDS (that is, it is visible to multiple logical partitions (LPARs) on your machine), then to be on the safe side we recommend initializing and labelling all the Linux DASD with a volume serial that makes it obvious that these are Linux disks. That way, if one or more of the disks are inadvertently varied online to another LPAR on the machine, it should be obvious that the disk belongs to the Linux environment.

A further tip would be to spread the minidisks of each Linux guest across physical volumes.This increases the I/O bandwidth available to each guest. It also avoids the situation where a process is blocked from reading in data so Linux tries to swap in another process, only to find that blocked because the swap file is on that same physical device. Having all minidisks the same size makes it easy to spread them over multiple physical volumes.

## 4.2.3 Virtual disks

Virtual disks are disks emulated by VM in memory. The z/VM operating system automatically manages the virtual disks. Typically they are defined in the guest user’s directory entry by address, along with the memory to allocate for them.

It is very important to remember that the contents of a virtual disk will *not* be available after a logoff from the virtual machine, or after a shutdown and IPL of the z/VM system. The typical usage of virtual disks for Linux will be as swap disk. Domino can also make use of virtual disks, for example as a place to temporarily store the data for rebuilding the database indices while running the Update task.

## 4.3 Brief introduction to Linux and UNIX filesystem

In this section, we discuss some differences between Linux/UNIX and Windows filesystems. Linux is a variant of UNIX, so those familiar with Linux or UNIX experts can skip this section and go to “Domino filesystem structure” on page 66.

The Linux filesystem differs from the filesystem used by Windows or DOS in several ways. For people that are used to the DOS filesystem, these differences may cause problems. The major differences are:

- Linux is case-sensitive
- Linux uses the forward slash (/) instead of the backslash (\) as a delimiter
- Linux uses a tree hierarchy instead of “drive letters”
- The same things are called by different names (for example, what is known as a “directory” in Linux is called a “folder” under Windows).
- Linux does not recognize the file type by file extensions (for example, `.exe` does not make an executable, and `.txt` does not link the file to an editor).

### 4.3.1 Linux is case-sensitive

“Case sensitivity” means that an upper case name is treated differently from the same name in lower case. So the following file names will refer to *different* files in Linux, but to the same file in Windows (when in the same directory or folder):

- file1.text
- File1.TEXT
- FILE1.text

### 4.3.2 Backslash versus forward slash

The backslash (\), which is used to separate directories under Windows, is not supported for this function in Linux. As is the case with Web addresses inside your Web browser, Linux uses the forward slash in the path names between directories. The Notes client can continue to refer to databases on the server with a path containing a backslash. It is not necessary to change the Location document on the Notes client to use forward slashes in pathnames.

### 4.3.3 The directory tree

One of the major differences between the Linux and the Windows filesystem is the lack of “drive letters”, which simply do not exist in the Linux world. Instead of using letters such as `C:\`, `D:\`, and so on, Linux places every directory in a tree (or root-) structure with one starting point. This starting point is known as root (/).

You can expand a Linux filesystem by mounting new partitions or drives onto a directory somewhere in this tree structure. However, directories cannot span mount points. To expand a directory, a new subdirectory can be added at a mount point, or an LVM can be expanded or defined to host the directory. An example of this tree structure is shown in Figure 4-6 on page 73.

#### Current working directory

In Linux, the directory you are currently in is called the *current working directory*. You can always display your current working directory with the `pwd` UNIX command:

```
linuxc: # pwd
```

```
/domservc/notesdata
```

## 4.3.4 Pathname

A *pathname* is the address for a specific file. Pathnames describe the location of a file. There two flavors of pathnames: absolute pathnames, and relative pathnames.

### Absolute pathname

An absolute path begins with a leading forward slash (/) to indicate the start from the root directory. For example, /domserva/notesdata/log.nsf is an absolute path to the log database located in the notesdata directory under the domserva under root directory.

### Relative pathname

Path names without a leading / are called relative paths and are relative to the current working directory. If the current directory is /domserva/notesdata, then the relative path mail1/user1.nsf refers to the file with an absolute path of /domserva/notesdata/mail1/user1.nsf.

When listing the contents of a directory, you may notice two special directory names: a single dot (.) and a double dot (. .), which have the following meanings:

- A single dot (.) indicates the current working directory.
- Double dots (. .) refer to the directory one level up.

Therefore, ./notes.ini refers to /domserva/notesdata/notes.ini, if the current working directory is still /domserva/notesdata.

## 4.3.5 Symbolic links

*Symbolic links* are pointers to a file or directory, which allow you to have multiple references of a file in multiple directories and only one physical representation of that file. Using symbolic links allows you to easily share and control information (files) among directories or users without copying the file to the different locations.

**Note:** We do not recommend the use of Linux symbolic links for Domino-related files. Instead, use Domino database and directory links.

## 4.3.6 File permissions

In Linux there are three basic attributes for file permissions: read, write, and execute. These attributes are known as *permission bits* and are represented as `r`, `w` and `x`. Three clusters of those permission bits are assigned to every file in a Linux filesystem. The clusters describe the file permissions for the file owner or user, the owning group, and the rest of the world.

The read permission bit grants read access to a file or directory, and the write bit allows write access to the file. For files, the execute bit allows the execution of those files. For directories, it allows you to search the directory even if you are not allowed to read the directory. Table 4-3 lists these file permission bits.

*Table 4-3 UNIX file permission bits*

| user | group | world (others) |
|---|---|---|
| rwx | rwx | rwx |

| user | group | world (others) |
|---|---|---|
| 421 | 421 | 421 |

The permission bits have also a numerical representation; in this representation, r=4, w=2, and x=1. To describe the permission for a user, you add up all the numbers. So 7 allows full access, 6 allows read and write, and 5 allows for read and execute.

The owner of the file has the first set of permission bits, then the group of the file, and then anyone not in this group. The following example shows the permission bits, the owner, and group setting for files beginning with: `no` in the data directory of our Domino server.

```
root@linuxa:/domserva/notesdata > ll no*
-r--r--r--    1 domserva notes     1049088 Apr  6  2000 nornyn.dic
-rw-r-----    1 domserva notes        1704 Aug 14 11:21 notes.ini
-rw-r-----    1 domserva notes        1692 Aug 14 11:20 notes.ini.1
-rw-r-----    1 domserva notes        1692 Aug 14 11:19 notes.ini.2
```

As you can see:

- Everyone in the system is allowed to read the nornyn.dic file, but no one is allowed to alter it.
- Members of the notes group and the domserva user are allowed to read the notes.ini file, but only domserva can change the file.

Table 4-4 lists combinations of file permission bits.

*Table 4-4 Permission bits and their meaning*

| Permission bits | Allow access to | Remarks |
|---|---|---|
| rwxrwxrwx | Everyone can read, write and execute this file. | Not recommended. |
| rwxrwxr-x | The owner and owner's group can read, write and execute this file. Everyone else can read and run it. | Setting for system-wide programs and scripts. |
| rwxr-x--- | The owner has full access, the group has read and execute rights, but all others are not allowed to access the file. | Only the owner can alter the file. |
| rw-rw-rw- | Everyone can read and write this file | This is a common setting for any data file that must be modified system-wide. |
| rw-rw-r-- | The owner and group can read and write, but all others may only read. | |

Table 4-5 on page 60 lists important, typical Domino files with their recommended permission bit settings.

*Table 4-5 Domino files and their permission bits*

| File name | Permission | Allowed to access | Remarks |
|---|---|---|---|
| notesdata | drwxr-x--- | read, search access for owner and group write access for owner | The “home” directory of the Domino server; note the leading “d” to mark a directory |
| notes.ini | -rw-r--r-- | read access for everyone write access for owner (server) | The configuration file for the Domino server |
| names.nsf | -rw------- | read ,write access for owner only | The Domino Directory definition database for Notes users and the Domino server |
| log.nsf | -rw------- | read ,write access for owner only | The logging database for the Domino server |
| mail or application databases | -rw------- | read ,write access for owner only | The actual data databases of the Domino server |
| mail or application directory | drwxr-x--- | read, search access for owner and group write access for owner | The directories for Domino application and mail databases |

### 4.3.7 Filesystem types: Journaled or non-journaled

There are multiple types of filesystems available under Linux for zSeries. There is ext2, a non-journaled filesystem that we used in some of our test implementations of Linux for zSeries running Domino. Filesystems like ext3, JFS, and Reiser are journaled filesystems that are equally valid for running Domino and have been used in testing Domino for Linux for zSeries at IBM.

The difference between a journaled and non-journaled filesystem is that a journaled filesystem offers rapid restart capability after a system crash. For example, if you are using ext2 (no journaling) and Linux crashes, then the filesystem may not be able to be mounted until it is consistency-checked using the **`e2fsck`** command. ext2 can take days to recover 1 terabyte of data after a system crash (power outage or re-IPL Linux under z/VM) without a graceful shutdown.

In contrast, with one of the journaled filesystems, changes to the filesystem are replayed from a checkpoint in the journal in much the same way as Domino transaction logging replays transactions against databases after a Domino crash. This can greatly reduce the amount of time that a server is down when a Linux crash has occurred. Therefore, many Linux experts recommend choosing ext3 or one of the other journaled filesystems.

However, everything has its price. The journaling under Linux requires a certain amount of overhead. This overhead may affect Domino server performance. This said, the benefits that are gained from running a journaled filesystem far outweigh the costs. Although all journaled filesystems have been tested, we used ext3 because we think it has less overhead than the others.

For further information about the ext3 filesystem, see:

http://www-106.ibm.com/developerworks/Linux/library/l-fs7/
http://www-106.ibm.com/developerworks/Linux/library/l-fs8/

## 4.4 Logical Volume Manager (LVM)

**Note:** This section contains text derived from *Configuring Logical Volume Management (LVM) on Linux for zSeries*, TIPS0128, and from the following site:

http://www.gweep.net/~sfoskett/Linux/lvmlinux.html

Disks cannot span more than one physical DASD volume. Therefore, for most applications, you have to combine several physical volumes into one larger logical volume.

When considering the layout of a filesystem to hold Domino data, it is often desirable to logically group databases with a common use in the same directory. On a mail server, you may want break up 1,000 mail users hosted on a server into five directories of 200 mail databases each. On an application server, you may create a separate directory for each of a number of applications.

In either case, these subdirectories may be very large, far exceeding the size of a single DASD volume. Without a volume management system, this limits the size of a filesystem on Linux for zSeries to the size of a DASD volume; for a 3390 model 3, this translates to 2.3 GB of available space for a Linux filesystem. Volume management systems allow multiple physical DASD devices to be combined into a single logical volume which can be used for a Linux filesystem, thereby overcoming the single DASD volume size limit.

In this section, we describe the concept of Logical Volume Manager (LWM), and how to configure it for Linux on zSeries. But first, let's review the special LVM terminology:

- A DASD volume is called a *physical volume* (PV), because that is the volume where the data is physically stored.
- The PV is divided into several *physical extents* (PEs) of the same size. The PEs are like blocks on the PV.
- Several PVs make up a *volume group* (VG), which becomes a pool of PEs available for the logical volume (LV).

The LVs appear as normal devices in /dev/ directory. You can add or delete PVs to or from a VG, and increase or decrease your LVs. The connections between the single elements defined above are shown in Figure 4-2 on page 62.

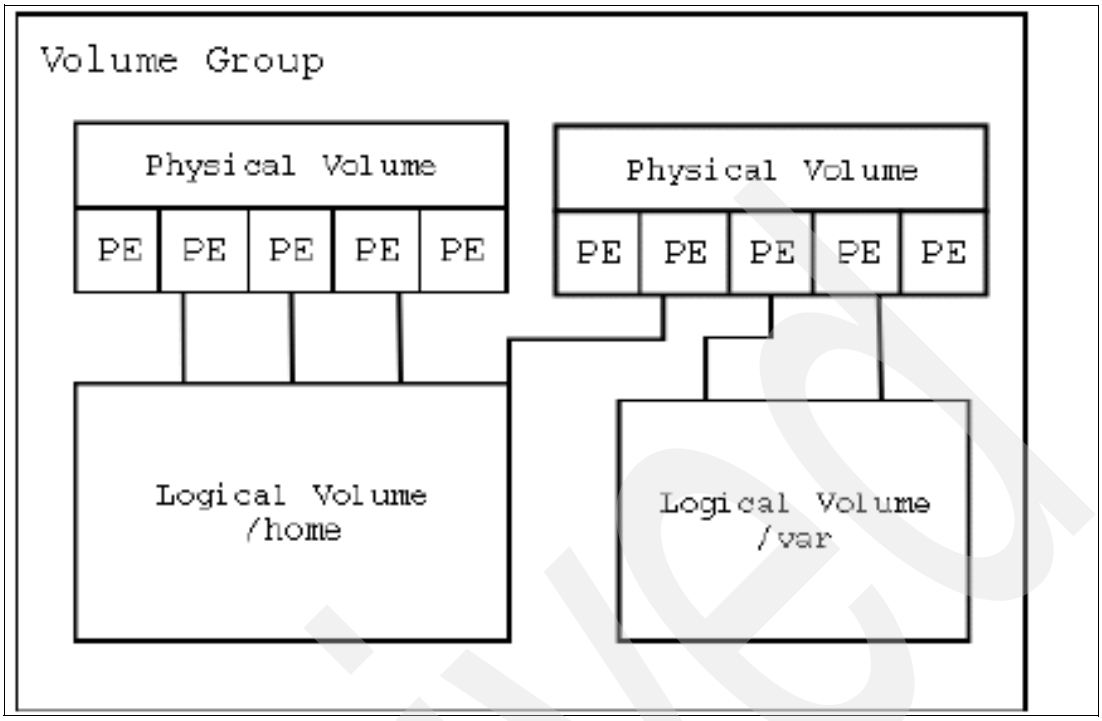



*Figure 4-2 LVM elements*

Logical Volume Manager provides a fundamental way to manage UNIX storage systems in a scalable, forward-thinking manner. Implementations of the logical volume management concept are available for most UNIX operating systems. These often differ greatly, but all are based on the same fundamental goals and assumptions.

Logical Volume Manager (LVM) abstracts disk devices into pools of storage space called volume groups (VGs). These volume groups may then be subdivided into virtual disks called logical volumes (LVs). These may be used just like regular disks, with filesystems created on them and mounted in the UNIX filesystem tree.

As mentioned, there are many different implementations of logical volume management. One, created by the Open Software Foundation (OSF), was integrated into many UNIX operating systems, and it also serves as a base for the Linux implementation of LVM that is covered here. Note that many other vendors offer logical volume management that is substantially different from the OSF LVM presented here (for example, Sun ships an LVM from Veritas with its Solaris system).

### Benefits of logical volume management

Logical volume management provides benefits in the areas of disk management and scalability.

**Note:** It is not intended to provide fault-tolerance or extraordinary performance (for this reason, it is often run in conjunction with RAID, which can provide both of those benefits).

By creating virtual pools of space, an administrator can assign chunks of space based on the needs of a system's users. For instance, an administrator can create dozens of small filesystems for different projects and add space to them as needed with very little disruption.

When a project ends, the administrator can remove the space and return it into the pool of free space—and even create a logical volume and filesystem that spans multiple disks.

(Contrast this with just “slicing up” a hard disk into partitions and placing filesystems on them; they cannot be resized, nor can they span disks.)

#### Costs of logical volume management

Logical volume management does extract a penalty because of the complexity and system overhead it occurs; it adds an additional logical layer or two between the storage device and the applications.

#### Volume groups

A volume group should be thought of as a pool of small chunks of available storage. It is made up of one or more physical volumes (partitions or whole disks, called PVs). When it is created, it is divided into a number of same-size chunks called physical extents (PEs). A volume group must be contain at least one entire physical volume, but other volumes may be added and removed in real-time as needed.

#### Logical volumes

Logical volumes are virtual disk devices made up of logical extents (LEs). LEs are abstract chunks of storage mapped by the LVM to physical extents in a volume group. A logical volume must always contain at least one LE, but more can be added and removed in real-time.

### 4.4.1 LVM for Linux

The OSF LVM was implemented on Linux and is now extremely usable and full-featured. There is a home page for this Linux LVM implementation. This LVM is quite similar to the LVM found on HP/UX, Digital, and AIX. It serves as an excellent model and “sandbox” for learning about LVM on those platforms. LVM will probably be integrated into future Linux kernels, but for now it must be added manually.

### 4.4.2 How to use LVM

LVM can be exercised either from the command line, or by using the setup tool YaST. Note, however, that if the YaST implementation does not complete successfully, your new logical volume could be left in an inconsistent state that may be difficult to correct. Therefore, we list the manual LVM steps.

- Creating physical volumes for LVM
- Registering physical volumes
- Creating a volume group
- Creating a logical volume
- Creating a filesystem on the logical volume
- Adding a disk to the volume group
- Creating a striped logical volume
- Moving data within a volume group
- Removing a logical volume from a volume group
- Removing a disk from the volume group

We give more detail in 6.5.9, “Set up logical volumes” on page 121.

### 4.4.3 Our example

These are the steps we used to create the filesystem for one of our Linux systems. During the workload tests, we had to rearrange some of the directories on our DASD, so this example shows you one way to create LVM and directories.

```
root@linuxa:/ > vgscan
```

```
vgscan -- reading all physical volumes (this may take a while...)
vgscan -- "/etc/lvmtab" and "/etc/lvmtab.d" successfully created
vgscan -- WARNING: This program does not do a VGDA backup of your volume group

root@linuxa:/ > cd domserva/notesdata

root@linuxa:/domserva/notesdata > df .
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/dasdb1            2259188   1101640   1157548  49% /

root@linuxa:/domserva/notesdata > ll
total 0
drwxr-xr-x    5 domserva notes         120 Jul 31 19:08 .
drwxr-xr-x    3 domserva notes          80 Jul 31 19:08 ..
drwxr-xr-x    2 domserva notes          48 Jul 31 19:08 mail1
drwxr-xr-x    2 domserva notes          48 Jul 31 19:08 mail2
drwxr-xr-x    2 domserva notes          48 Jul 31 19:08 translog

root@linuxa:/domserva/notesdata > df
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/dasdb1            2259188   1101640   1157548  49% /
/dev/dasdc1            2403184   1735800    667384  73% /opt
shmfs                   257140         0    257140   0% /dev/shm

root@linuxa:/domserva/notesdata > pvcreate dasd[f-l]1
pvcreate -- invalid physical volume name "dasd[f-l]1"

pvcreate [-d|--debug] [-f[f]|--force [--force]] [-h|--help]
        [-s|--size PhysicalVolumeSize[kKmMgGtT]] [-y|--yes] [-v|--verbose]
        [--version] PhysicalVolume [PhysicalVolume...]

root@linuxa:/domserva/notesdata > for i in d e f g h i j k l m n o p q r s t u
> do
> fdasd -a /dev/dasd$i
> done
auto-creating one partition for the whole disk...
writing volume label...
writing VTOC...
rereading partition table...
...
auto-creating one partition for the whole disk...
writing volume label...
writing VTOC...
rereading partition table...

root@linuxa:/domserva/notesdata > pvcreate /dev/dasd[f-l]1
pvcreate -- physical volume "/dev/dasdf1" successfully created
pvcreate -- physical volume "/dev/dasdg1" successfully created
pvcreate -- physical volume "/dev/dasdh1" successfully created
pvcreate -- physical volume "/dev/dasdi1" successfully created
pvcreate -- physical volume "/dev/dasdj1" successfully created
pvcreate -- physical volume "/dev/dasdk1" successfully created
pvcreate -- physical volume "/dev/dasdl1" successfully created

root@linuxa:/domserva/notesdata > pvscan
pvscan -- reading all physical volumes (this may take a while...)
pvscan -- inactive PV "/dev/dasdf1" is in no VG  [2.29 GB]
pvscan -- inactive PV "/dev/dasdg1" is in no VG  [2.29 GB]
pvscan -- inactive PV "/dev/dasdh1" is in no VG  [2.29 GB]
pvscan -- inactive PV "/dev/dasdi1" is in no VG  [2.29 GB]
```

```
pvscan -- inactive PV "/dev/dasdj1" is in no VG  [2.29 GB]
pvscan -- inactive PV "/dev/dasdk1" is in no VG  [2.29 GB]
pvscan -- inactive PV "/dev/dasdl1" is in no VG  [2.29 GB]
pvscan -- total: 7 [16.04 GB] / in use: 0 [0] / in no VG: 7 [16.04 GB]

root@linuxa:/domserva/notesdata > vgcreate mail1 /dev/dasd[f-l]1
vgcreate -- INFO: using default physical extent size 4 MB
vgcreate -- INFO: maximum logical volume size is 255.99 Gigabyte
vgcreate -- doing automatic backup of volume group "mail1"
vgcreate -- volume group "mail1" successfully created and activated

root@linuxa:/domserva/notesdata > vgdisplay mail1
--- Volume group ---
VG Name               mail1
VG Access             read/write
VG Status             available/resizable
VG #                  0
MAX LV                256
Cur LV                0
Open LV               0
MAX LV Size           255.99 GB
Max PV                256
Cur PV                7
Act PV                7
VG Size               16 GB
PE Size               4 MB
Total PE              4095
Alloc PE / Size       0 / 0
Free  PE / Size       4095 / 16 GB
VG UUID               lUDs5e-5ROE-mo6C-dMR8-YLS1-jKpq-rOzS7s

root@linuxa:/domserva/notesdata > lvcreate --extents 4095 -n mail1 /dev/mail1
lvcreate -- doing automatic backup of "mail1"
lvcreate -- logical volume "/dev/mail1/mail1" successfully created

root@linuxa:/domserva/notesdata > mke2fs /dev/mail1/mail1
mke2fs 1.28 (31-Aug-2002)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
2097152 inodes, 4193280 blocks
209664 blocks (5.00%) reserved for the super user
First data block=0
128 block groups
32768 blocks per group, 32768 fragments per group
16384 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
        4096000

Writing inode tables: done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 37 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
root@linuxa:/domserva/notesdata > cp /etc/fstab /etc/fstab.org
root@linuxa:/domserva/notesdata > vi /etc/fstab

/dev/dasdb1          /                    reiserfs   defaults              1 1
```

```
/dev/dasdc1          /opt                 reiserfs   defaults              1 2
/dev/dasda1          swap                 swap       pri=42                0 0
/dev/mail1/mail1     /domserva/notesdata/mail1 ext2  defaults              0 0
devpts               /dev/pts             devpts     mode=0620,gid=5       0 0
proc                 /proc                proc       defaults              0 0

root@linuxa:/domserva/notesdata > mount /domserva/notesdata/mail1
root@linuxa:/domserva/notesdata > df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/dasdb1           2.2G  1.1G  1.2G  49% /
/dev/dasdc1           2.3G  1.7G  652M  73% /opt
shmfs                 252M     0  252M   0% /dev/shm
/dev/mail1/mail1       16G   20K   15G   1% /domserva/notesdata/mail1
```

## 4.5 Domino filesystem structure

To understand where to place the files of your Domino server, you have to understand which files the server uses and what they are good for. By default, all files are placed under the notesdata directory—some in dedicated directories, and others directly in notesdata.

The most important files for a Domino server are the ID file (server.id), the configuration file (notes.ini), and the Domino Directory (names.nsf). Those files are typically placed directly in the notesdata directory. Figure 4-3 shows a typical directory structure of a Domino server.

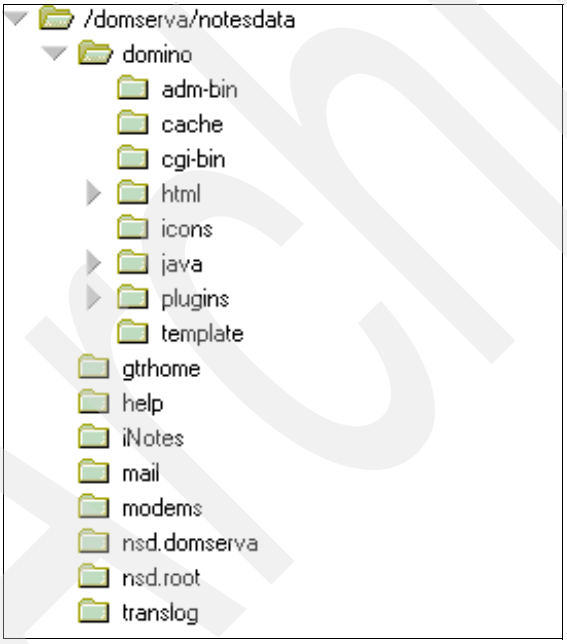



*Figure 4-3 Domino directory structure*

For the files that hold the application and mail data, you should use dedicated directories, filesystems, and mount points. Figure 4-3 shows this for the mail directory. From the start it is a good idea to have the notesdata directory, the mail directory, and the translog directory as separate mount points with separate volumes for each attached. In the following sections, we discuss in more detail the placement of files and databases. Of course, not only the data files go into the notesdata directory and below; all files related to the code (executables, libraries, and so on) go under the /opt/lotus directory, by default.

### 4.5.1 Mount points and Domino

**Tip:** Whenever a mount point is created for Domino, the owner and group of the mount point must be changed to be the correct owner and group for that mount point. For example, if the mount point is to be used by “domserva” and is part of the group “notes,” then execute the following:

```
chown -R domserva:notes <mount point>
```

Proper planning of the mount points can have a significant effect on the manageability of Linux and the Domino server. For Domino, we recommend you create at least three filesystems or mount points for the Domino server: one for the binaries (/opt/lotus), a second for the notesdata directory (.../notesdata), and a third for the directory holding the actual users or application databases (.../mai01 or .../apps01). If you use Domino transaction logging you should plan to have a separate filesystem and mount point for the /translog directory. As discussed later in this chapter, place this /translog directory on disk with high performance and low contention.

Normally you will want to have additional mount points so that you can structure your Domino data in more detail. Figure 4-4 and Figure 4-5 on page 68 illustrate the relationship of mount points to the directories in two types of Domino servers. The directories in bold and italics indicate that these are separate mount points.

Figure 4-4 shows an example of the structure of the Domino filesystem. We do not show all the directories that might occur in a productive Domino server. The figures focus on the mount points and on the naming convention for the directories.

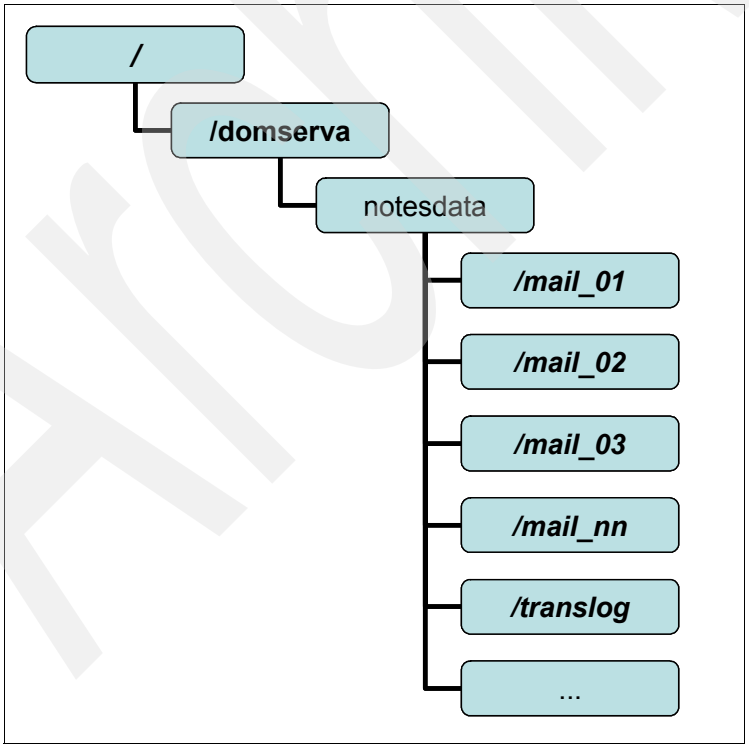



*Figure 4-4 Example mount point (filesystem ) structure for a mail server*

The directory domserva is the first mount point for Domino. The notesdata directory is a directory under domserva and within the domserva mount point. The directories mail_01,

mail_02, mail_03, mail_nn are also separate mount points, as is translog. Other directories can exist under notesdata, in addition to the mail_xx and translog directories.

For an application server, the structure might look more like Figure 4-5. Before making decisions on where to place your mount points, read the following sections about Domino databases and Domino-related files and where they can be placed.

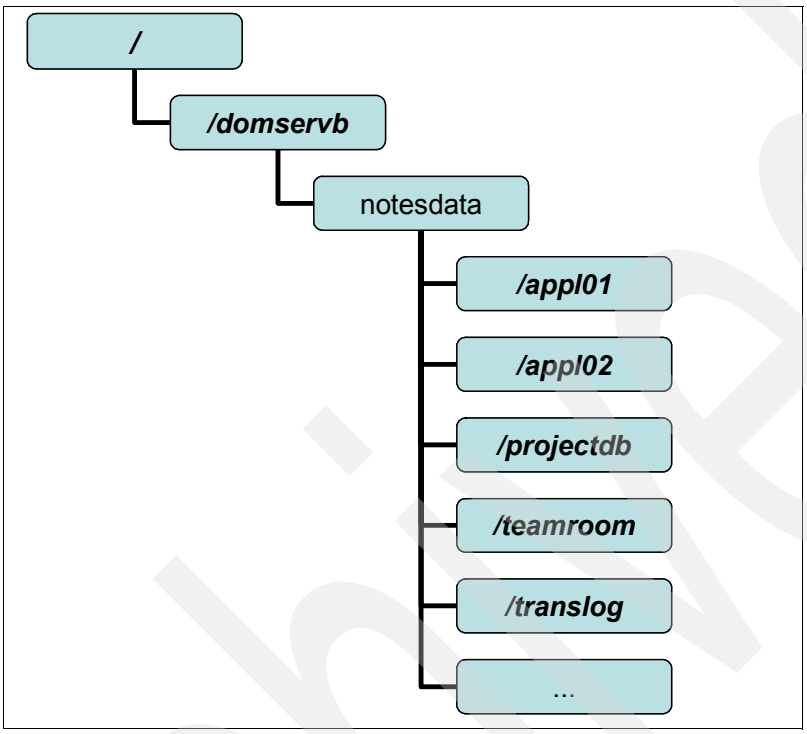



*Figure 4-5 Example mount point structure for an application server*

## 4.6 Placement of high-use Domino files

In this section we discuss the placement of important Domino databases and files. We point out how the structure of your filesystem can influence the overall performance of your Domino server.

### 4.6.1 Transaction log

When transaction logging is enabled, the speed at which transactions are written to the log files has a direct impact on server performance. When a transaction (update to a database by an NRPC call) is closed, it is immediately committed to the transaction log. When the transaction has been fully committed to the log, it is marked as “complete”. Shortly after this, the changes to the database are built in memory and eventually written to the actual database on physical disk. Because of this sequence, the performance of the device which houses the transaction log determines the overall performance of the server.

The transaction log is made up of a number of 64 megabyte transaction log extent files. These files are written to the log sequentially and almost never read from.

The sequential writing of data ensures that the motion of the heads across the disk surface will be in constant direction with little or no extraneous head movement, hence there is no disk latency. This ensures the absolute best performance, even on older disk subsystems. The

disk subsystem must also be able to handle continuous writes at a speed of at least 10 MB/sec.

It is an absolute necessity that the transaction log directory *not* be placed on a disk with other high-use data, as access to these files would degrade the overall performance of the transaction log. Place the transaction log on a dedicated device and a low-use controller if possible. This recommendation is due to the requirement of being able to deliver a constant stream of transactions to the disk subsystem at the fastest possible transfer rate.

An additional consideration is the fact that the transaction log is now the most important data file on the system. This is especially true if archive style transaction logging is being used. If a database or group of databases need to be restored, they can be brought up to absolute currency by restoring the database from the last full backup, and then applying the transaction logs. If the databases are completely lost due to a disaster, they can always be recovered as long as the transaction log extents are fully intact.

Keeping this discussion in mind, it is best practice to place the transaction log directory on its own mount point. This mount point should be configured on the best performing DASD. Multiple full minidisks or DASD volumes in an LVM should be configured to house the transaction log. The DASD string should, if possible, be associated with channels that are lightly loaded in order to achieve the best throughput.

We recommend that you define a separate (mount point) filesystem for the transaction logs. Together with your DASD administrator, try to make sure that this filesystem is fast and has enough space left for your preferred transaction log operation (archive or circular). If you already have a Domino infrastructure, you can use the server statistics to calculate how many transactions during a day are written by your servers. Also, from the server statistics, find the typical size of the transactions. With both numbers, you can easily set up a filesystem that is large enough to support your Domino environment.

**Tip:** Enable the fast write cache on your DASD

Fast write cache (FWC) is an optional, nonvolatile cache that provides redundancy with the standard adapter cache. The FWC tracks writes that have not been committed to disk.

FWC can significantly improve the response time for write operations. However, care must be taken not to flood the cache with write requests faster than the rate at which the cache can destage its data. It can also adversely affect the maximum I/O rate, because additional processing is required in the adapter card to determine if the data that is being transferred is in the cache.

Fast write cache typically provides significant advantages in specialized workloads (for example, copying a database onto a new set of disks). If the fast write cache is spread over multiple adapters, this can multiply the benefit.

The FWC can also reduce journaling filesystem (JFS) log bottlenecks because of three properties of the JFS log:

- The JFS log is write-intensive. The FWC does not cache unmodified data.
- The writes are small and frequent. Because the cache capacity is not enormous, it works best for high-rate small I/Os that are gathered together in the adapter into larger physical I/Os. Larger I/Os tend to have better performance because less disk rotations are normally needed to write the data.
- Logs are not typically very large, relative to the cache size, so the log does not tend to “wash” the cache frequently. Therefore, the log loses the benefit of rewriting over existing cached data.

### 4.6.2 Names.nsf

In most cases, you will accept the default location for the Domino Directory (formerly known as name and address book or NAB), which is the notesdata directory. When grouping multiple volumes together into one logical volume, it is not necessary to have a separate filesystem for the Domino Directory. However, there may be times when it is a good idea to configure a mount point with a dedicated minidisk or volume to hold the Domino Directory.

These may include a full-text indexed Domino Directory with frequent indexing, a directory that is being used as a repository for user information (such as an online company phone and address book), or any other situation where the directory is being heavily accessed. If you move the Domino Directory, you will need a database link for it, because by default the server expects the Domino Directory to be in the notesdata directory.

### 4.6.3 mail.boxes

On mail servers, some of the most heavily used and most dynamically changeable databases are the mail.box databases. In most cases, placing the mail.box database within notesdata provides adequate performance to meet most needs. In rare cases, if mail performance is shown to need improvement, the mailx.box databases can be placed in a separate filesystem with their own mount point. Remember to include a database link in notesdata.

**Tip:** Make sure transaction logging is disabled for your mail.boxes

### 4.6.4 Log files for the Java Console (dcntrlr*)

Put the log files in a separate directory. This is not a performance hint, but doing this helps you to clean up your notesdata directory. The log files for a remote Domino Java™ console can be configured with an expiration date and therefore be controlled.

### 4.6.5 Indexing and other temporary files

For performance reasons, we suggest that you place the temporary files for rebuilding the index of a database on a virtual disk. You can force Domino to use a place other than default by setting the following notes.ini variable: `View_Rebuild_Index_Dir=your/directory.` For the same reason, you can put the directory that your anti-virus software uses for scanning and decompressing attachments on a virtual disk. Also you might think of placing the gtrhome directory on a virtual disk. gtrhome is the directory in which the index engine stores its temporary results from queries, and where it builds temporary indexes when users full text search databases that do not have indexes (usually through one of the APIs or an agent).

> **Attention:** Be sure to put only *non-permanent data* onto a virtual disk! It will be lost when the user ID is logged off or the server is shut down.

### 4.6.6 Use of database and directory links

Both Linux and Domino offer symbolic links for directories and files to databases. We do not recommend the use of Linux symbolic links. But the deployment of Domino database or directory links can help you structure your environment without struggling with the Notes client settings.

For example, you can direct all mail users to a directory called mail. Inside this directory, the database links redirect the user transparently to the physical location of their mail file. To accomplish this, you have to create a database link for every mail database in your system. When deleting or adding a user, remember to delete or create not only the database, but also the link.

## 4.7 Placement of other Domino databases

This section focuses on the data files of your Domino server. The preceding sections described where to put the files and databases related to the Domino server system. The following sections describe what to keep in mind when dealing with the user or application databases.

### 4.7.1 Mail files

The mail databases should go into a separate filesystem below the notesdata directory of the Domino server. To determine how many mail directories you need in your environment, look at the number of mail users, the total size of the mail databases (including fulltext index), and the growth-rate of your mailfiles (define quota is always a good idea).

Another point is how long it takes for a full recovery of one directory; you can simply put all your mail databases into one directory by defining a huge filesystem, but what will your users say if you take 24 hours to recover the directory?

From a performance point of view, it is not important to spread the databases among multiple filesystems. Because of the virtual device concept in VM, Linux has no effect on which physical device the data is written to. If you run your Linux natively on a zSeries machine, you must take care of your device-to-filesystem allocation.

### 4.7.2 Application databases

What is true for the placement and filesystem considerations for mail databases is also true for your application databases; make sure you have a separate filesystem set up for your application data, and that it is large enough to hold all the data. For applications, that might not only be Notes databases and full-text indices, but also flat files that these applications import or export.

### 4.7.3 Estimating DASD space

The amount of DASD space that is needed for Domino depends on the amount of data that you will store. This is the same as for other Domino platforms. For mail databases, this is normally in the range of 50 to 150 megabytes per user, and will typically grow as the users become more advanced in their use of Domino. Installations with thousands of users will therefore require hundreds of gigabytes, if not terabytes, of DASD space.

Before installing Linux, you should also plan for the proper sizing of the root filesystem and the /opt filesystem. The binaries for products installed will typically be placed in a subdirectory under /opt. The Domino binaries need about 1.3 GB. For our installation, an additional 600 MB was needed for the tar file. The size of the tar file may be larger for the release you are installing.

A good starting point might be to size the root (/) and /opt filesystems at about 5 GB or two 3390 mod-3 full pack minidisks. root (/) and /opt should be defined as separate mount points and not combined into a single filesystem. Also plan for /swap space, although it is far better to use virtual disk rather than real DASD, at least in a z/VM environment. This provides significant performance enhancements.

**Lesson learned:** During testing, we failed to take into account the size of the /domserva and notesdata directories, and placed them under root (/). We found that we had to move the notesdata into a separate filesystem. This caused a considerable amount of work, which could have been avoided had we planned better.

### 4.7.4 Naming convention

We recommend that you use a self-describing, self-documenting naming convention for filesystems and LVMs within your DASD configuration.

At the Linux level, name your LVMs so that the purpose of the LVM is obvious. We used a volume group name of mail1 and a logical group name of mail1 for the LVM used for the .../notesdata/mail1 directory (and mount point). Another lesson learned was that there were problems that were not obvious at first glance when we created a volume group for all of Domino and the individual logical groups for the subdirectories. It is also a good idea to have a self-describing naming convention for your directories.

At the z/VM level, you cannot assign a name or label to a minidisk, because the minidisk definition just maps a logical device address to a physical device. When assigning DASD to z/VM, it is a good idea to have a sequence of addresses to belong to the same z/VM LPAR. The VOLSERs for these DASD packs should indicate their use on Linux systems. As an

example, we used a convention of LXxxxx, where xxxx was the device address for the VOLSERs of packs used under Linux.

For example, as you can see in Figure 4-4 on page 67, our notesdata directory is located under the domserva directory, which describes the Domino server it belongs to. We planned it this way so that, if we were to add an additional Domino partitioned server (DPAR) to the LinuxA virtual machine, we would already have a self-documented structure in place: domserva with its notesdata directory below it for DomServA/ITSO, and in the future, domservx with its notesdata directory for DomServX/ITSO.

### 4.7.5 Example of our filesystem

Under VM, we installed four Domino servers in three separate Linux VM guests. So we had two Domino servers in their own Linux and two partitioned Domino servers in a third Linux system. In Figure 4-6, the filesystem for a Linux hosting one Domino server is shown. Only the directories that begin with a /(in bold and italics) are mount points. For clarity we have omitted other (unimportant) directories inside /opt and notesdata.

The notesdata directory is a directory under domserva and within the domserva mount point. The directories mail_01, mail_02, mail_03, mail_nn are also separate mount points, as is translog.

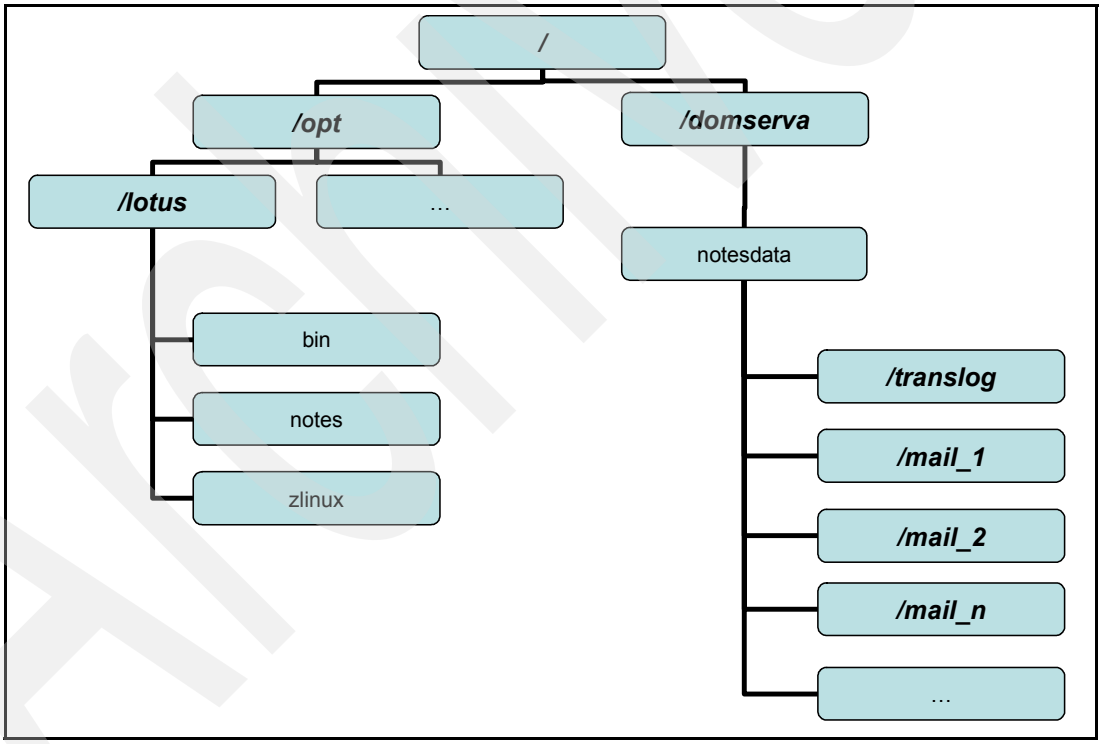



*Figure 4-6 Our filesystem structure*

Much thought went into deciding on the structure for the servers used in this redbook project. There is a greater view that must be looked at when planning the use of DASD than we show in the example. All of the servers that will run in the z/VM LPAR or the Linux LPAR as a whole should be considered and planned for. Look at the total view of all DASD used across all servers.

In our case, we debated for a long time about the best way to lay out our mail and other database directories. In doing so, we considered the total size of mail files, total number of

databases, the channel path to the DASD, and the types of DASD available to use (ESS versus RVA), and so on. All of these factors must be looked at before considering how to lay out the Domino directory structure.

Even before installing Linux, a decision had to be made as to which of the filesystem types to use, journaled or non-journaled. For the journaled filesystems, we had a choice of types. In at least one instance, we used ext2, a non-journaled filesystem. For Domino, we found this is a mistake. It can lead to very long startup times for the Linux guest in the event of a crash. ext2 also performs extensive checks on the filesystem after every twenty starts of the Linux image.

In retrospect, we should have used ext3 or another journaled filesystem to improve reliability of the filesystems on all Linux images. A journaled filesystem is highly recommended for use with Domino. ext3 has been part of the Domino product test in the Poughkeepsie Lab, as well as during our redbook project.

In Chapter 2, “Planning” on page 13, we provide a worksheet to help you plan the use of the mount points and the filesystems. At the very least, plan for separate filesystems at mount points for the directories listed in Table 4-6.

*Table 4-6 Domino filesystems*

| Directory | Use |
|---|---|
| / | root |
| /opt | product code - Depending on the number of products installed in the Linux image, it may be best to create separate mount points for specific installed products. |
| /lotus (under /opt) | Domino Server binaries and scripts. A separate filesystem at its own mount point can ease backing off of an upgrade, and so on. |
| /tmp | temporary - May be highly variable in size. |
| /notesdata | Domino server data directory. While the directory for the servername can be under root (that is, domserva) the notesdata should be under its own mount point. |
| /mailxx or /appsxx | The individual application and mail database directories should planned based on size of databases, recovery consideration, DASD type, and so on. |
| /translog | Definitely in its own filesystem |
| /swap | Linux swapping - Use virtual disk when possible. |

5

# Networking

In this chapter, we discuss networking issues for a Domino server on Linux running on zSeries hardware, usually under z/VM. We address two models (OSA card and z/VM Guest LAN), as well as other networking issues specifically related to Domino.

The chapter is divided into the following sections:

- z/VM networking
- Linux networking
- Domino networking
- Troubleshooting your network

## 5.1 Two common networking methods

Two common ways to connect Linux images in zSeries over a TCP/IP network are:

- Directly over a shared OSA adapter
- Over a z/VM guest LAN with TCP/IP as a router

Figure 5-1 shows a directly-connected shared OSA on the left side, and a z/VM Guest LAN with a z/VM TCP/IP router on the right side.

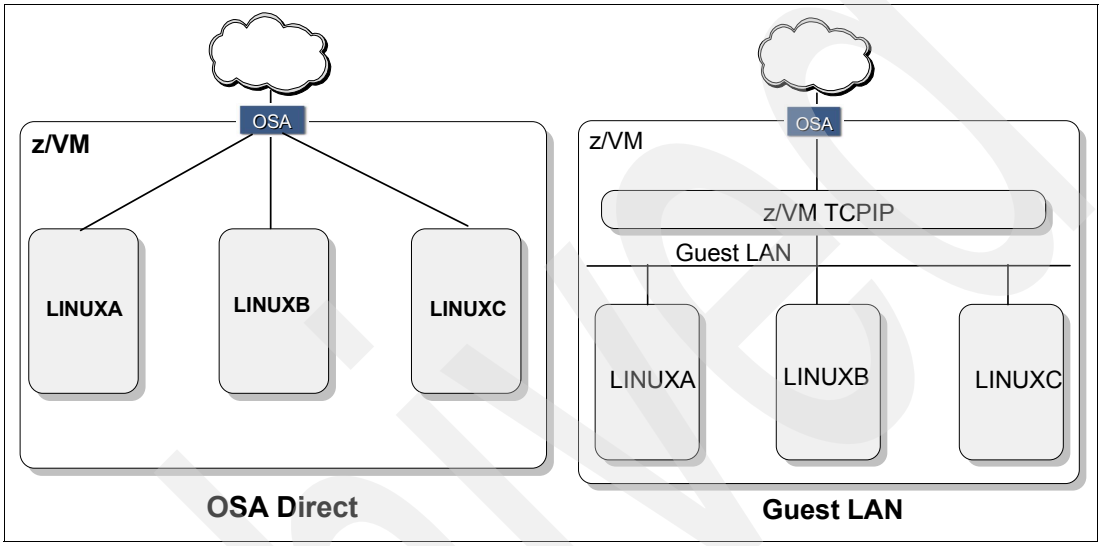



*Figure 5-1 Two common methods of networking Linux on zSeries*

**Important:** A virtual switch (VSWITCH) function has been introduced with z/VM 4.4. It looks like the Guest LAN shown on the right-hand side, but avoids the cost of running a virtual router and simplifies the routing.

## 5.2 z/VM networking

This section describes the minimum tasks needed to connect your z/VM system to the network. For more details, refer to *TCP/IP Planning and Customization*, SC24-6019.

There are two z/VM user IDs that provide basic TCP/IP services.

**TCPIP** The TCPIP user ID provides the primary TCP/IP service, called the *stack*. When it is logged on or disconnected, TCP/IP should be running. When it is not logged on, TCP/IP is down.

**TCPMAINT** Owns TCP/IP production resources—the 198, 591, and 592 disks. It is the user ID traditionally used to maintain the TCP/IP environment in z/VM.

At a minimum, the following configuration files must be updated to connect the z/VM TCP/IP to the network:

**TCPIP DATA** The TCPIP DATA file defines system parameters used by TCP/IP client applications. It is used to specify configuration information such as the host name of the z/VM host, the user ID of the TCPIP virtual machine, the domain name, and name servers. This

information is mostly similar to what you find on Linux in the file /etc/resolv.conf.

**PROFILE TCPIP** The PROFILE TCPIP file contains the information required by TCP/IP to connect to the network. Minimally, one pair of DEVICE and LINK statements must be defined. The link is assigned a home address. The network routing environment is established with static routes and a default route. Finally, the device has to be started. On a Linux system, this information would be found in several places: /etc/route.conf, /etc/modules.conf, /etc/inetd.conf, or /etc/sysconfig/network/ifcfg-eth0.

**SYSTEM DTCPARMS** The file can be used to separate system programming and network programming responsibilities. The sections that follow do not address the use of this file. Rather, all definitions that can be configured in the file are placed in the USER DIRECT or TCPIP PROFILE files.

**Tip:** It is often helpful to run two TCP/IP stacks on z/VM. It is relatively easy to do with a few changes to configuration files. Then you can continue to run on the system when you take one of the stacks down. If you have more than one TCP/IP stack, it is a good idea to separate production and test systems to simplify dealing with changes in both.

In the following sections, we discuss some basic networking tasks that are commonly done in a z/VM and Linux environment:

- Give OSA addresses to Linux user IDs
- Add a guest LAN to the system
- Add NICs to Linux user IDs
- Customize TCP/IP
- Start or stop TCP/IP
- Dynamically changing TCP/IP

### 5.2.1 Give OSA addresses to Linux user IDs

In order for z/VM to give OSA addresses to Linux user IDs, it must have OSA addresses defined to it. OSA CHPIDs are defined in the system IOCDS. They can be defined as either shared or reconfigurable. If you want more than one LPAR to be able to use the OSA at the same time, you must define the CHPID as shared. If only a single LPAR needs the OSA, then it is your choice which you use. In the world of OSD CHPIDs, since the definition of IP addresses is no longer “burned into the card” (that is, manually defining and installing the OSA address table (OAT) into the card), sharing a card is greatly simplified.

For example, to define the OSAs for the network connectivity on this project, CHPID 00 is defined as an OSD CHPID in SHR mode. Devices 2C00 through 2C0F are defined (2C00-2C0E are OSA, and 2C0F is OASD). CHPID 03 is defined as an OSD CHPID in SHR mode. Devices 2C60 through 2C6F are defined (2C60-2C6E are OSA, and 2C6F is OASD).

You can query the OSA addresses that are being used with the **QUERY OSA** command, and the OSA addresses that are available with the **QUERY OSA FREE** command. For example:

```
q osa
OSA  2C00 ATTACHED TO TCPIP    2C00
OSA  2C01 ATTACHED TO TCPIP    2C01
OSA  2C02 ATTACHED TO TCPIP    2C02
OSA  2C04 ATTACHED TO LINUXA   2C04
OSA  2C05 ATTACHED TO LINUXA   2C05
```

```
OSA  2C06 ATTACHED TO LINUXA   2C06
OSA  2C08 ATTACHED TO LINUXB   2C08
OSA  2C09 ATTACHED TO LINUXB   2C09
OSA  2C0A ATTACHED TO LINUXB   2C0A
q osa free
OSA  2C03 FREE    , OSA  2C07 FREE    , OSA  2C0B FREE    , OSA  2C0C FREE
OSA  2C0D FREE    , OSA  2C0E FREE    , OSA  2C0F FREE    , OSA  2C20 FREE
OSA  2C21 FREE    , OSA  2C22 FREE    , OSA  2C23 FREE    , OSA  2C24 FREE
OSA  2C25 FREE    , OSA  2C26 FREE    , OSA  2C27 FREE    , OSA  2C28 FREE
...
```

**Tip:** This setup skips one device per OSA triplet (that is, the first triplet is 2C00, 2C01, 2C02, and the second triplet is 2C04, 2C05, 2C06). This model will limit each OSA to 60 TCP/IP stacks (240/4). Some older qeth drivers require the first address to be even. To get to the 80 stacks per OSA, the addresses can be assigned 00, 01, 02 to the first, and 04, 05, 03 to the second, 06, 07, 08 to the 3rd, 0a, 0b, 09 to the fourth, and so on.

Once the z/VM LPAR has OSA addresses, the **DEDICATE** statement can be used in the user directory entry to “give” the addresses to a Linux user ID. For example, the following statements in the USER DIRECT file under the definition of a user ID will give it two OSA triplets that will support two TCP/IP addresses:

```
DEDICATE 2C60 2C60
DEDICATE 2C61 2C61
DEDICATE 2C62 2C62
DEDICATE 2C63 2C63
DEDICATE 2C64 2C64
DEDICATE 2C65 2C65
```

If the OSA addresses are given to a Linux user ID in the USER DIRECT file and that user ID is currently logged on, be sure to shut down Linux and log off and back on again, in order to allow the changes to take effect.

If you want to allocate OSA addresses interactively, the ATTACH command can be used to give an OSA triplet (the three ports necessary to support communication) to Linux user IDs that are logged on. For example, to give LINUXB the OSA triplet at address 2C08, 2C09, and 2C0A, the following command can be used:

```
att 2c08-2c0a linuxb
2C08-2C0A ATTACHED TO LINUXB
```

The channel device layer in Linux allows for this device to be used without shutting down.

**Tip:** All of our examples keep the virtual and real addresses the same. Another method is to give each Linux the OSA devices at the same virtual address, then map them in the directory. This allows you to move things around without impacting the individual Linux guest.

### 5.2.2 Add a guest LAN to the system

A z/VM Guest LAN provides a method of creating virtual networks. Figure 5-2 on page 79 shows three Linux guests that have both direct OSA connections and z/VM guest LAN connections. Note which aspects of the network are physical and which are virtual.

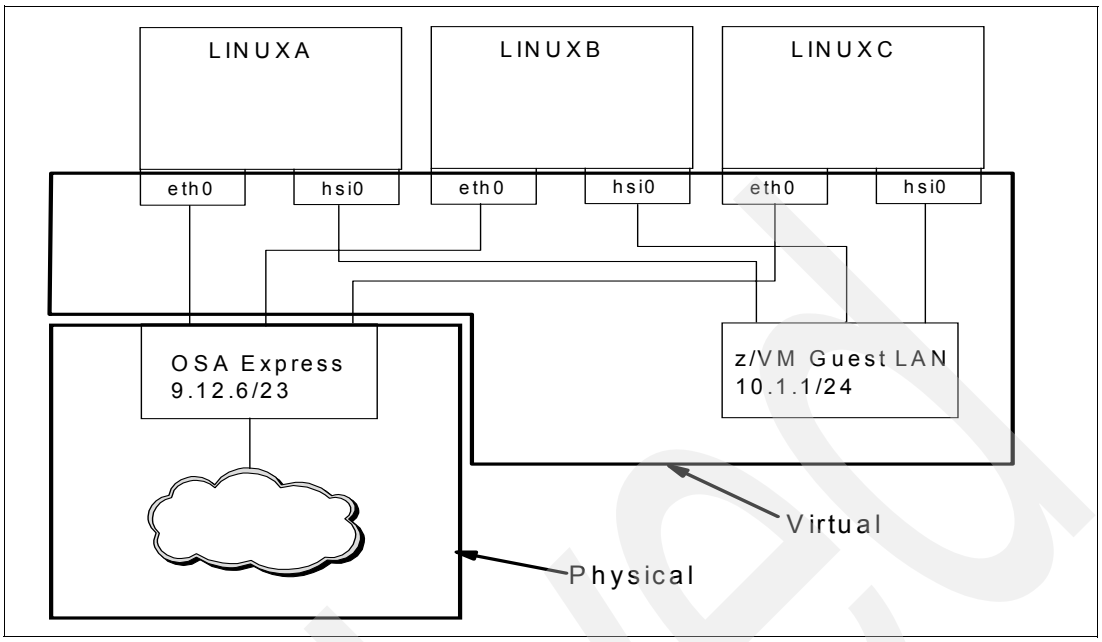



*Figure 5-2 Sample Guest LAN configuration*

Guest LANs can be one of two types: HiperSockets (the default) or QDIO. A HiperSocket guest LAN is recommended because it might have a slight performance advantage over a QDIO guest LAN, and also because the interface names on Linux will have a prefix of “hsi”, which might help distinguish them from physical interfaces (if a QDIO guest LAN is used, both physical ethernet OSAs and virtual guest LAN interfaces will have a prefix of “eth”).

**Tip:** Having the same name (for example, eth0) for the network interface, whether you have a shared OSA device or a Guest LAN adapter, could be an advantage because you will not have to change the definitions inside Linux when you move things.

To define a Guest LAN, it is recommended you define the LAN in two ways:

- Interactively, on the command line, and
- Permanently, in the SYSTEM CONFIG file

The interactive definition is good for the existing z/VM session, while the definition in the SYSTEM CONFIG file ensures the same LAN will be defined after a re-IPL. In both cases, the syntax is the same. Following is an example of defining a HiperSocket guest LAN named `GESTLAN1` via the command line on the MAINT user ID. After the LAN is created, it can be queried with the **`QUERY LAN`** command:

```
DEFINE LAN GESTLAN1 MAXCONN 100 OWNERID SYSTEM
LAN SYSTEM GESTLAN1 is created
q lan
LAN SYSTEM GESTLAN1    Type: HIPERS   Active: 0     MAXCONN: 100
  PERSISTENT  UNRESTRICTED  MFS: 16384   ACCOUNTING: OFF
```

The same line should be added to the `SYSTEM CONFIG` file. To update it, you will probably have to release it as a CP disk using the **`CPRELEASE`** command. You can then link to the disk read/write (MR) and access it; for example:

```
q cpdisk
Label  Userid   Vdev Mode Stat Vol-ID Rdev Type   StartLoc     EndLoc
MNTCF1 MAINT    0CF1  A   R/O  430RES 0200 CKD         278        306
```

```
MNTCF2 MAINT    OCF2  B   R/O  430RES 0200 CKD         307        335
MNTCF3 MAINT    OCF3  C   R/O  430RES 0200 CKD         336        364
cprelease a
CPRELEASE request for disk A scheduled.
HCPZAC6730I CPRELEASE request for disk A completed.
link * cf1 cf1 mr
DASD OCF1 LINKED R/W; R/O BY     2 USERS
acc cf1 f
```

You now have the SYSTEM CONFIG file read/write on your F disk. Back it up and add the **DEFINE LAN** statement to the end of the file. After modifying the file, it is prudent to check the syntax using the **CPSYNTAX** command which is on MAINT's 193 disk; for example:

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

Then you can bring the MAINT CF1 disk back online via the **CPACCESS** command:

```
rel f
det cf1
DASD OCF1 DETACHED
cpaccess maint cf1 a
CPACCESS request for mode A scheduled.
HCPZAC6732I CPACCESS request for MAINT's OCF1 in mode A completed.
```

The next time z/VM is IPLed, the guest LAN will be defined from the SYSTEM CONFIG file.

### 5.2.3 Add network interface cards (NICs) to Linux user IDs

After you have defined the Guest LAN, you need to define virtual network interface cards (NICs) for each Linux image. This can be done interactively using the **DEFINE NIC** command. However, it is recommended that you define the NIC and couple it automatically by an entry in the USER DIRECT file, or by a directory maintenance product, if you are using one. This is done through the user directory **SPECIAL** statement.

For example, to define a NIC at addresses 600, 601, and 602 that can connect to the system HiperSocket LAN named GESTLAN1, add the following statement to the USER DIRECT file for each user that will need one. If you are using a PROFILE for Linux user IDs, add the statement to the PROFILE so that the Guest LAN triplet will be defined to all Linux user IDs:

```
SPECIAL 600 HIPER 3 SYSTEM GESTLAN1
```

After changing the USER DIRECT, remember to run **DIRECTXA** to bring your changes online. Also, if any of the Linux user IDs are logged on at the time, be sure each one logs off and back on again so the change can take effect.

If you are using DIRMAINT, the command would be similar. For example, to add a NIC to the user ID LINUXA, the command would be:

```
DIRM FOR LINUXA SPECIAL 600 QDIO 3 SYSTEM GESTLAN1
```

### 5.2.4 Customize TCP/IP

To bring z/VM onto the network, TCP/IP must be customized. This is not complicated, though some network administrators feel the syntax of the configuration files is unusual compared with other systems. The two main configuration files are the TCPIP DATA and the PROFILE TCPIP files. The IFCONFIG command can also be used to configure network interfaces for the z/VM TCP/IP stack, or to display the current configuration. IFCONFIG does not make

permanent changes to the network configuration, but it can provide data for this purpose that is compatible with the TCP/IP server configuration file.

## Customizing the TCPIP DATA file

Log on to TCPMAINT. The TCPIP DATA file is on the TCPMAINT 198 minidisk, which should be accessed as your D disk. XEDIT the file and make the following changes.

Define the DOMAINORIGIN which is the Domain Name Services (DNS) suffix. For example, our suffix is itso.company.com:

```
DOMAINORIGIN  itso.company.com
```

Define the DNS name server. For example, our server is at IP address 184.156.23.7:

```
NSINTERADDR  184.156.23.7
```

## Customizing the PROFILE TCPIP

The PROFILE TCPIP file is also on the TCPMAINT 198 minidisk, which should be accessed as your D disk. The following sections or statements are needed in the PROFILE TCPIP file to put z/VM on the network:

- A DEVICE and LINK statement
- A HOME address for the link
- A GATEWAY and DEFAULTNET statement
- A START statement

Now we describe the important statements in the file, and give an example of how our TCPIP virtual machine is attached to the network; it uses an OSA express connection.

### *A DEVICE and a LINK statement*

Add two lines to the file and enter the following DEVICE and LINK statements:

```
DEVICE OSA2C00D OSD 2C00 PORTNAME OSA2C00  PRIROUTER
LINK OSA2C00L     QDIOETHERNET  OSA2C00D
```

This defines the device named OSA2C00D as an OSD. Addresses 2C00, 2C01, and 2C02 are the three addresses of the OSA CHPID that are used. OSA2C00 is the PORTNAME. The LINK statement defines a link named OSA2C00L. The protocol to be used is queued direct I/O (QDIO) protocol over ethernet. This link is associated with the device OSA2C00D. The PRIROUTER parameter specifies that all packets be routed to the VM TCP/IP stack, except those intended for other directly connected stacks. This allows further routing to all hosts on the guest LAN. There can be only one stack on the OSA as primary router.

### *A HOME address*

The HOME statement simply provides a home TCP/IP address for the defined link1:

```
184.156.23.82    OSA2C00L
```

Just as you associate an IP address with a network device, such as eth0, eth1 in UNIX/Linux, you also use the HOME statement to associate an IP address to LINK name (in this example, OSA2C00L).

### *GATEWAY statement*

Add the following GATEWAY statement to define the 10.28.29 network:

```
; (IP) Network  First        Link      Max. Packet  Subnet      Subnet
; Address       Hop          Name      Size (MTU)   Mask        Value
; -----------   ------------ -------   -----------  ----------- --------
   9             =            OSA2C00L 1500         0.255.254.0 0.12.6.0
```

This statement adds a network address of 9, which is accessible through the link named OSA2C00L. The maximum transmission unit (MTU) size is 1500. Defining the subnet mask in this fashion is probably different from what you are used to doing. The subnet mask for this 10 network definition is 0.255.254.0, and the subnet values are 0.12.6.0. The subnet mask is 255.255.254.0.

> **Tip:** A "gotcha" when configuring the TCPIP PROFILE is that the GATEWAY statement must not go past column 72. For historical reasons, columns 73 to 80 cannot be used. This is easy to overlook, as the last set of dashes in the comment end on column 72 and not much space is allotted for that field. If your statement does go past column 72, the error message that results is not especially clear.

#### *DEFAULTNET statement*

Next, add the following DEFAULTNET statement:

```
DEFAULTNET  184.156.23.92      OSA2C00L  1492        0
```

This defines the default gateway as the host with TCP/IP address 184.156.23.92.

#### *START statement*

Finally, add a START statement for the device named OSA2C00D.

```
START OSA2C00D
```

This will start the OSA "device".

### 5.2.5 Start or stop TCPIP

The user ID named TCPIP is called a *service machine*. It is analogous to a Linux daemon; when a daemon is running, the service is available. When a z/VM service machine is logged on or disconnected, the service is available. Therefore, starting TCP/IP on z/VM is as simple as logging on the user ID TCPIP, and stopping it as simple as logging off.

#### Starting TCPIP manually

When you are first customizing the TCPIP user ID, it is recommended that you log onto it, press Enter when prompted, and read the messages as the TCP/IP stack begins. After you are sure it will start consistently, you can use the **XAUTOLOG** command; or better, you can have TCPIP start automatically.

#### Starting TCPIP automatically

You will probably want TCPIP to start automatically. To do this, log on to the AUTOLOG1 user ID and edit the PROFILE EXEC file. Add the following line:

```
ADDRESS COMMAND CP XAUTOLOG TCPIP
```

### 5.2.6 Dynamically changing TCP/IP

Having to bring TCP/IP up and down may be fine when your system is just being tested. However, when you have a production system, recycling TCP/IP can be a problem.

The **OBEYFILE** command allows you to make temporary dynamic changes to the system operation and network configuration without stopping and restarting the TCPIP virtual machine. You can maintain different files that contain a subset of the TCP/IP configuration statements and use **OBEYFILE** to activate them while TCP/IP is running; see "Configuring the TCP/IP Server" in *z/VM TCP/IP Planning and Customization,* SC24-6019.

## 5.3 Linux networking

A Linux guest can be installed, with the primary network interface being a guest LAN.

### 5.3.1 Connecting Linux to a z/VM guest LAN

If a guest LAN is defined and the z/VM Linux guests have a NIC defined and coupled (see 5.2.3, “Add network interface cards (NICs) to Linux user IDs” on page 80), a network interface to a Linux guest can be added as follows.

1. Update the file /etc/chandev.conf to define the device.
2. Update the file /etc/modules.conf to associate a driver to an interface.
3. Create a configuration file in the directory /etc/sysconfig/network/, as shown:

```
# cd /etc
# tail -1 chandev.conf
noauto;hsi0,0x0600,0x0601,0x0602;add_parms,0x10,0x0600,0x0602,portname:GESTLAN1
# tail -1 modules.conf
alias hsi0 qeth
# cd sysconfig/network/
# cat ifcfg-hsi0
BOOTPROTO="static"
STARTMODE="onboot"
IPADDR="10.0.0.1"
MTU="1500"
NETMASK="255.255.255.0"
NETWORK="255.255.254.0"
BROADCAST="255.255.254.0"
```

Figure 5-3 on page 84 illustrates possible Domino, Linux, and VM configurations that might be considered, depending on your environment.

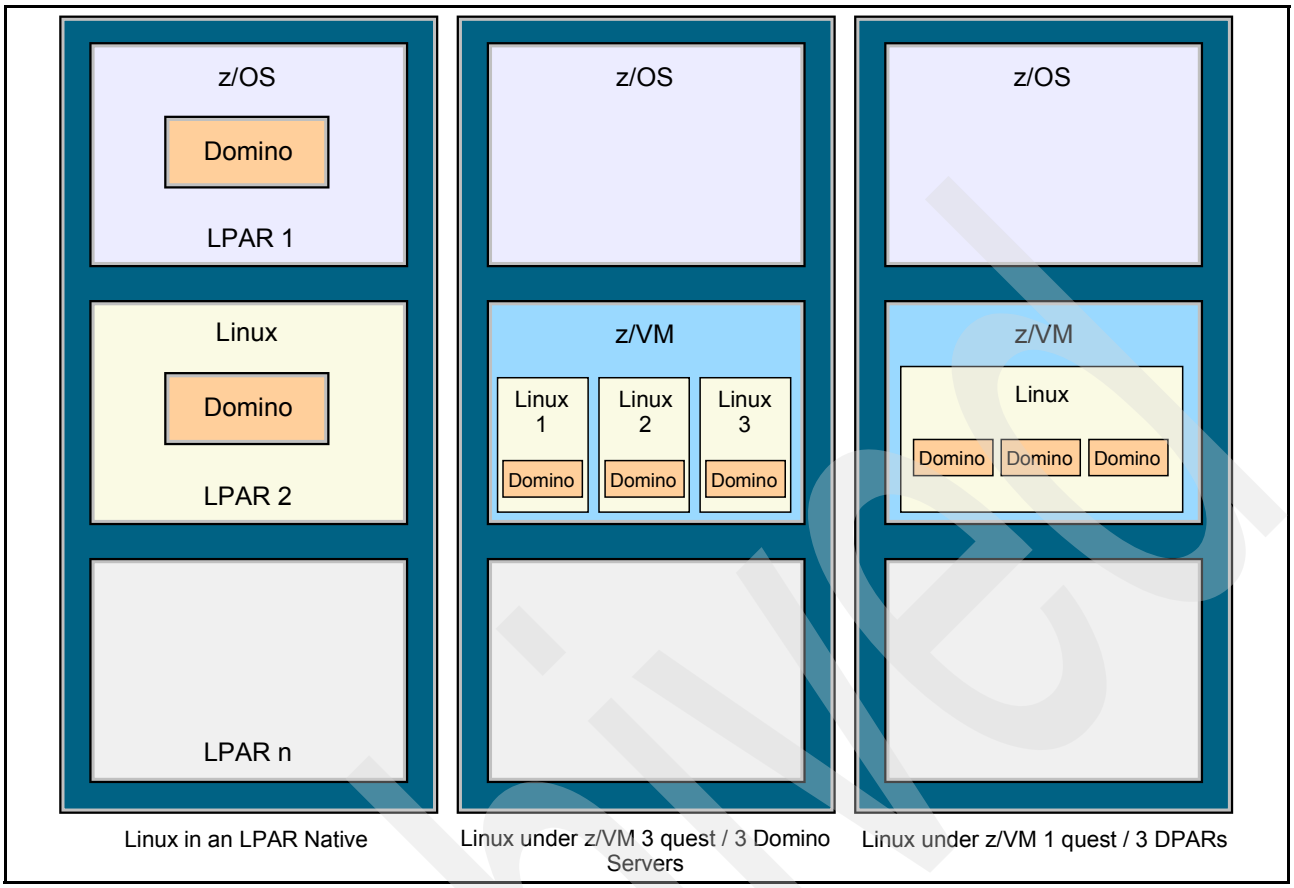



*Figure 5-3 Ways to configure Domino servers on Linux*

## 5.4 Domino networking

The functionality of Notes workstations, Internet access, and Domino servers depends on the effectiveness and capacity of networks. To plan a Domino network with sufficient capacity, you must consider not only the traffic to and from Domino servers, but also any other traffic on the network.

### 5.4.1 Notes remote procedure call (NRPC) communication

Domino servers offer many different services. The foundation for communication between Notes workstations and Domino servers or between two Domino servers is the Notes remote procedure call (NRPC) service.

### 5.4.2 Notes network ports

During the server setup program, Domino provides a list of Notes network ports based on the current operating system configuration. If these ports are not the ones you want to enable for use with the Domino server, you can edit the list during setup.

## 5.4.3 Notes named networks (NNN)

Consider Notes named networks (NNN) in your planning. A Notes named network is a group of servers that can connect to each other directly through a common LAN protocol and network pathway—for example, servers running on TCP/IP in one location. Servers on the same NNN route mail to each another automatically, whereas you need a Connection document to route mail between servers on different NNNs.

Figure 5-4 shows an NNN example. The servers are in the same domain and use the same network protocol.

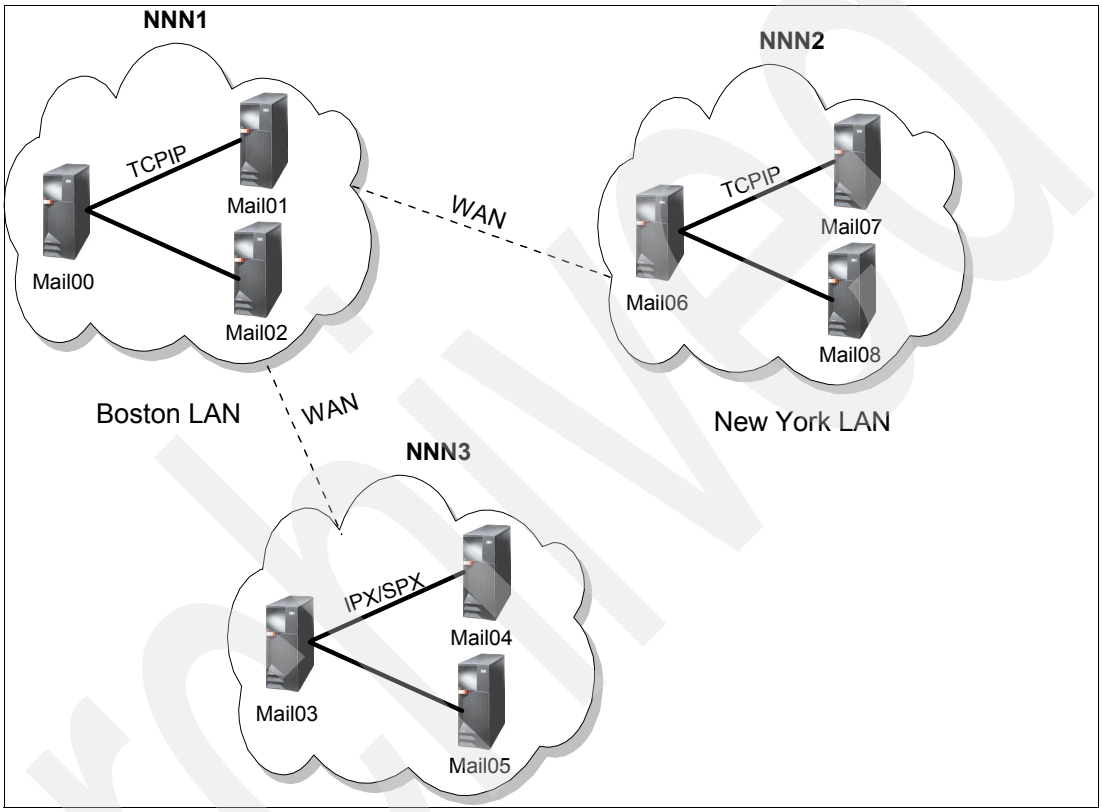



*Figure 5-4 Notes named networks servers*

**Important:** Domino assumes that all servers in a NNN have a continuous LAN or WAN connection. If this is not the case, serious delays in mail routing between servers can occur. Be careful not to include servers with only dial-up connections in an NNN.

When you set up server documents, be sure to assign each server to the correct NNN. Lotus Domino expects a continuous connection between servers that are in the same NNN, and serious delays in routing can occur if a server must dial up a remote LAN because the remote server is inadvertently placed within the NNN. Also bear in mind that the Notes Network field for each port can contain only one NNN name, and no two NNN names can be the same.

NNNs affect Notes users when they use the Open Database dialog box. When a user selects **Other** to display a list of servers, the servers displayed are those on the NNN of the user's home server for the port on which the Notes workstation communicates with the home server. Also, when users click a database link or document link, if a server in their home server's NNN has a replica of that database, they can connect to the replica.

**Note:** If a server is assigned to two NNNs in the same protocol, as in the case where the server has two Notes network ports for TCP/IP, a Notes workstation or Domino server connecting to that server uses the NNN for the port listed first in the Server document.

### 5.4.4 Resolving server names to network addresses in NRPC

Communications between Lotus Notes and Lotus Domino run over the NRPC protocol on top of each supported LAN protocol. When a Notes workstation or Domino server attempts to connect to a Domino server over a LAN, it uses a combination of the built-in Notes Name Service and the network protocol's name-resolver service to convert the name of the Domino server to a physical address on the network.

The Notes Name Service resolves Domino common names to their respective protocol-specific names. Because the Notes Name Service resolves common names by making calls to the Domino Directory, the service becomes available to the Notes workstation only after the workstation has successfully connected to its home (messaging) server for the first time. (The protocol name-resolver service normally makes the first connection possible.)

When the Notes workstation makes a subsequent attempt to connect to a Domino server, the Notes Name Service supplies it with the Domino server's protocol-specific name—that is, the name that the server is known by in the protocol's name service, which is stored in the protocol's Net Address field in the Server document. The protocol's name-resolver service then resolves the protocol-specific name to its protocol-specific address, and the workstation is able to connect to the server.

**Note:** When resolving names of Domino servers that offer Internet services, Lotus Notes uses the protocol's name-resolver service directly.

### 5.4.5 TCP/IP: Domino on Linux

**Note:** This information can also be found in the Release Notes for Domino 6.5.

Because the TCP/IP protocol is built into the Linux operating systems, you do not need any additional network software to set up a Domino server and TCP/IP on a Linux system. If you use DNS or a local host file, ping the Domino server by IP address and by host name. Then install the Domino server and run the server setup program.

There is an issue with the default size of the network queue as set on Linux with the kernel parameter tcp_max_syn_backlog. The default value is 5, but under certain conditions, that may not be large enough. On other UNIX systems the default is 20, which is a better setting.

It is recommended that the user change this parameter in their kernel by changing the value stored in /proc/sys/ipv4/tcp_max_syn_backlog (/proc/sys/ipv6/tcp_max_syn_backlog for ipv6) such that it is no less than 20. Since this is a parameter in the running kernel, it must be done each time the system is booted. Easy ways to accomplish this are by making an entry in your startup scripts (such as rc.local), or by making a change in sysctl.conf.

### 5.4.6 Recommendations for Domino partitioned servers (DPARs)

1. Use IP addresses *only* when configuring hostname for partitioned servers.
2. Set up partitioned servers using separate TCP/IP addresses.

   See the Domino Administration online help topic; select **LAN Configuration** -> **Partitioned servers** -> **Partitioned servers and the TCP/IP network**. To assign separate IP addresses to partitioned servers, use the notes.ini variable:

```
TCPIP_TcpIpAddress=0,ipaddress
```

To bind port 80 to each partitioned server's HTTP process, as described in the online help, enable Bind to host name in the server document (**Internet Protocols tab** -> **HTTP tab**) for each server. Do *not* use the DNS hostname for this field; add each server using only the numeric IP address in the host name field.

Using a browser, verify that each partitioned server can respond to requests on Port 80, after restarting your server.

3. You can set up multiple Web sites using the new Domino/Notes 6 Web site document. On partitioned servers, you can set up virtual Web servers using server documents.

   When you assign IP addresses (hosts) to bind to a specific tap process, you must place the numeric IP addresses for each Web server in the hostname field of the server document. Do *not* use the DNS hostname for this field.

   For multiple addresses, separate them with semicolons (;). If you separate them by commas, they will be saved with semicolons; see Figure 5-5.

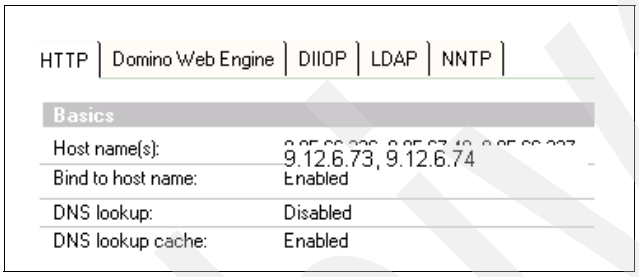



*Figure 5-5 Using IP addresses for Web servers*

   9.12.23.73 is the partitioned server. 9.12.23.73 and 9.12.23.74 are both Web sites (or virtual servers, if you are not using Web site documents).

   **Note:** You can have up to 32 entries in this field.

4. Create either Web Site Documents for the Web site, or virtual server documents to further define the HTTP configuration. There is a setting (Loads Internet Site Configuration Documents) in the Server Document on the first tab that must be checked to use only Web site documents to define the Web site.

5. Restart HTTP. You should now be able to send HTTP requests to the partitioned servers and all of the virtual servers or Web sites for each partition.

### 5.4.7 Network Address Translation (NAT)

NAT is a method of translating an IP address between two address spaces: a public space and a private space.

Public addresses are assigned to companies by the Internet Corporation of Assigned Names and Numbers (ICANN) or leased from the company's ISP/NSP. Public addresses are accessible through the Internet (routable) unless firewalls and isolated networks make them inaccessible.

Private addresses are IP address spaces that have been reserved for internal use. These addresses are not accessible over the Internet (non-routable), because network routers within the Internet will not allow access to them. The following address spaces have been reserved for internal use. It is best to use these IP addresses and not make up your own.

```
Class A: 10.0.0.0 to 10.255.255.255
Class B: 172.16.0.0 to 172.31.255.255
```

```
Class C: 192.168.0.0 to 192.168.255.255
```

For example, users inside a company access the Domino server based on its assigned IP address, which is a private address (192.168.1.1). Internet users must access the Domino server through a NAT router, which converts the private address to one of its static public addresses (130.20.2.2). Therefore, a Notes client accessing the server from the Internet uses the public address.

**Note:** Some companies use only public addresses. This avoids the potential duplication of private addresses if two companies were to merge their networks.

## 5.5 Troubleshooting your network

In this section we list some problem determination aids that may help to identify and correct network issues.

For Notes and Domino to work properly with TCP/IP, the protocol stack must be configured properly. In other words, Notes connectivity relies on TCP/IP communication.

**Note:** In all TCP/IP troubleshooting, first verify basic TCP/IP connectivity and configuration.

### Ping necessary addresses and names

The PING executable verifies IP-level connectivity. Example 5-1 shows some examples.

*Example 5-1 Examples of PING command*

```
prompt>  PING 1.1.5.15         (ping by address)
prompt>  PING Name             (ping by Server [possibly Alias] Name)
prompt>  PING Name.company.com (ping by FQDN Name)
```

If PING fails here, TCP/IP is not set up properly on the local machine. The site's network administrators should be contacted for technical assistance.

### Examine Notes server documents

In the server document of the Domino Directory, make sure the Notes server's common name in the TCP port's Net Address Field is the same name as the TCP Host name; see Example 5-2. If this is not the case, a name resolution alias is required in the HOSTS file or in the DNS table.

*Example 5-2 Naming the server*

```
Notes Server Name = Mail1
TCPIP Host Name = Mail1
Domain Name = company.com
TCPIP Fully Qualified Domain Name = Mail1.company.com
```

If any of these changes are made, you *must* restart the involved Domino server before changes to the Server Document will be enacted.

### Examine your firewall configuration

If you are having network problems, take a look at your firewall. Some default firewalls, even those by SuSE and Red Hat, are configured to protect a workstation rather than a server. This may cause your server firewall to hold back traffic for your TCP/IP applications.

## Other commands

Here are other commands you may find helpful when debugging network problems.

### *TRACEROUTE*

Issue the command TRACEROUTE to determine what lies between the source and the destination of IP traffic.

The TRACEROUTE command determines the route from one host to another through a network, and displays an ordered list of the routers in the path. There is a man page (Linux manual online) with the different parameters that you can use with this command.

### *NSLOOKUP*

This utility serves as the main diagnostic tool for DNS, and is supplied with the operating system. As with DNS, you must have TCP/IP installed before you can use it. NSLOOKUP is to DNS as PING is to general IP connectivity—the first tool you turn to for testing. It queries DNS name servers and shows you the results it receives. The command we used on our test systems was **`nslookup -sil`**.

### *NETSTAT*

Use NETSTAT to see what IP address is connecting to the server and on which port. The format will be **`<IPAddress>:<port>`** (for example, 10.10.10.10:1352). Use the command n**`etstat -an`**. To make reading the output easier, consider redirecting to an output file, for example:

```
netstat -an > netstat.txt
```

### *HOMETEST*

The HOMETEST command can test the system configuration including HOSTNAME, DOMAINORIGIN, and NSINTERADDR, which are defined in the TCPIP DATA file.

HOMETEST verifies that the host tables or name server (depending on the NSINTERADDR statement) can resolve the fully qualified domain name (defined by HOSTNAME and DOMAINORIGIN statements) for your site.

In addition, the Internet addresses corresponding to your site HOSTNAME are checked against the HOME list. This is defined in the PROFILE TCPIP file. A warning message is issued if any addresses are missing from the HOME list.

**Note:** Verify that the TCPIP virtual machine has been started before you use the HOMETEST statement.

6

# Linux installation

In this chapter, we discuss the installation of Linux for zSeries. Before you can install Domino, you first need at least one Linux image. Linux on zSeries can either be installed as a guest under z/VM, or directly on a zSeries LPAR. Installation of Linux both under z/VM and on an LPAR are discussed here.

## 6.1 Introduction

Prior to installing Linux and Domino on zSeries, you should undertake some planning as discussed in Chapter 2, “Planning” on page 13. Consider the personnel and skills that will be involved. Ideally there will be a z/VM person (if installing under z/VM), a Linux person, a network person, and a DASD person. Of course, in many organizations, individuals will perform multiple tasks.

### 6.1.1 Bill of materials

You will need the following items for your Linux installation:

- Either z/VM in an LPAR, or a zSeries LPAR for with adequate DASD and memory (refer to Chapter 4, “Disk configuration” on page 53).
- Linux distribution. Because of the need for the sys_epoll code, Domino can only run on United Linux 1.0 (SuSE SLES-8 or Turbolinux TLES-8 with service pack 2) at this time
- A Domino tar package.
- An NFS or FTP server to give access to the Linux distribution CDs.
- A workstation with an X server (optional; an X server is recommended but not required) and an ssh client (such as PuTTY). You might find it helpful to have a Linux workstation fill this role, for ease of copying and editing files.

### 6.1.2 High level steps

The high level steps involved in installing Linux and Domino are as follows:

1. Prepare for the Linux installation; refer to 6.2, “Prepare for Linux installation” on page 93.
2. Install and customize z/VM if Linux will be running under it; refer to 6.3, “z/VM first steps” on page 98.
3. If Linux will be running in an LPAR, refer to 6.4, “Prepare and IPL Linux in an LPAR” on page 99.
4. Install and customize Linux; refer to 6.5, “Linux installation and customization” on page 105.
5. Install Domino; refer to Chapter 7, “Domino installation” on page 133.

These steps are illustrated in Figure 6-1 on page 93.

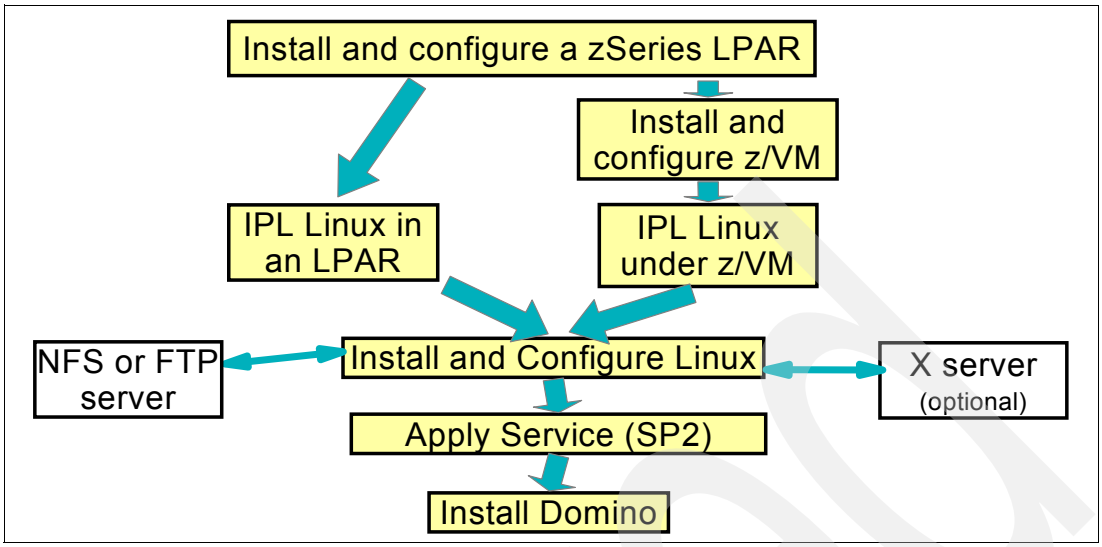



*Figure 6-1 Installation steps*

## 6.2 Prepare for Linux installation

The steps involved in preparing for a Linux installation are as follows:

- Obtain a Linux distribution.
- Obtain the necessary networking resources and information.
- Mount the Linux CDs and set up an NFS or FTP server.
- Fill in the Linux installation worksheet.

### 6.2.1 Obtain a Linux distribution

Table 6-1lists the 31-bit Linux distributions that are available for zSeries.

*Table 6-1 Linux for zSeries distributions*

| Distribution | Kernel level | Date of build (per uname -v) |
|---|---|---|
| Debian 3.0 | 2.4.17 | Dec. 16, 2002 |
| Red Hat 7.2 | 2.4.9 | Nov. 23, 2001 |
| SuSE Linux | 2.2.16 | Nov. 5, 2000 |
| SuSE SLES-7 | 2.4.7 | Oct. 30, 2001 |
| SuSE SLES-8 (United Linux 1.0) | 2.4.17 | Nov. 6, 2002 |
| SuSE SLES-8, with Service Pack 2 | 2.4.19 | June 5, 2003 |
| TurboLinux TLES-8 (UnitedLinux 1.0) | 2.4.17 | Oct. 30, 2001 |
| TurboLinux TLES-8 with Service Pack 2 | 2.4.19 | June 5, 2003 |

There are other Linux distributions for zSeries that are not commonly used, or are no longer available. They include Caiman Linux, Millenux Linux, and the Marist filesystem.

Domino for zSeries is not supported on any 64-bit Linux distributions.

**Important:** Domino 6.5 needs the new sys_epoll thread pooling code. Notes clients will not connect to the server if this code is not in place. Of all the distributions listed above, only United Linux 1.0 (31-bit SuSE SLES-8 and Turbolinux TLES-8) with Service Pack 2 has this function, at the time of writing.

The Web site for the SuSE SLES-8 zSeries distribution is:

http://www.suse.com/us/business/products/server/sles/s390.html

### 6.2.2 Obtain the necessary networking resources and information

You will need at least one networking device to install Linux on zSeries. Often this is a directly connected OSA or a z/VM guest LAN, both of which are usually defined by three zSeries addresses. For additional information about networking, see Chapter 5, “Networking” on page 75.

You will need at least one TCP/IP address and the usual related TCP/IP information, such as the gateway TCP/IP address, the DNS TCP/IP address, the subnet mask, and the MTU size.

You will need a host name for the Linux image. It is useful to have it in the Domain Name Services (DNS) ahead of time, but not required.

### 6.2.3 Mount the Linux CDs and set up an NFS or FTP server

Usually the Linux RPMs are copied from an NFS or FTP server. It can be a Linux on zSeries, however, you run into a “chicken and egg” problem: if you need a Linux FTP or NFS server to install zSeries Linux, then how do you install the first copy? For this reason, often a Linux distribution is installed on a PC first. If you are going to do this, it is recommended that you install the same distribution on a PC that you will be installing on zSeries. Then you have to make the CDs available by setting up either an NFS or FTP server.

**Attention:** An NFS server is recommended over an FTP server because it makes applying SLES-8 service pack 2 (sp2) easier—but either method will work.

#### Making the Linux CDs available

SuSE SLES-8 consists of three installation CDs, two supplemental CDs and service pack 2 (sp2) CD. You need at least installation CDs 1 and 2, and probably CD 3, to install Linux. The supplemental CDs are not required; however, some useful packages such as cpint are on them.

It is recommended that the physical CDs be copied to .iso files (also called *ISO images)*. This avoids the need to constantly have to change CDs in the drive. To convert physical CDs to ISO images, put each CD in the drive and use the **dd** command. For example:

```
# cd /sles8
# dd if=/dev/cdrom of=/SLES-8-s390-Int-RC6-CD1.iso
```

**Note**: Some Linux systems do not have a device file named /dev/cdrom, even though they have a physical CD drive. If so, you should determine the name of the CD drive.

When you have created ISO images for all necessary CDs, they must be mounted loopback. This is done with the **-o loopback** parameter to the **mount** command.

You will also want the filesystem to be mounted read-only, which is the nature of a CD, and you will want empty directories over which to mount (by convention, this is under the /mnt directory).

An example of mounting four ISO images loopback is shown here. After you are finished, verify that the CDs are mounted:

```
# cd /mnt
# mkdir sles8cd1 sles8cd2 sles8cd3 sles8sp2
# cd /sles8
# mount -o loop,ro /SLES-8-s390-Int-RC6-CD1.iso /mnt/sles8cd1
# mount -o loop,ro /SLES-8-s390-Int-RC6-CD2.iso /mnt/sles8cd2
# mount -o loop,ro /SLES-8-s390-Int-RC6-CD1.iso /mnt/sles8cd3
# mount -o loop,ro /SLES-8-s390-Int-RC6-CD1.iso /mnt/sles8cd1
# mount | grep sles8
/sles8/SLES-8-s390-Int-RC6-CD1.iso on /mnt/sles8cd1 type iso9660
/sles8/SLES-8-s390-Int-RC6-CD2.iso on /mnt/sles8cd2 type iso9660
/sles8/SLES-8-s390-Int-RC6-CD3.iso on /mnt/sles8cd3 type iso9660
/sles8/SLES-8-SP2-s390.iso on /mnt/sles8sp2cd1 type iso9660
```

The next step is to set up either an NFS (recommended) or FTP server.

## Set up an NFS server

In most SLES-8 installs, the NFS server is installed but not started. This is verified by the **rpm** and **rcnfsserver** commands:

```
# rpm -qa | grep nfs-utils
nfs-utils-1.0.1-25
# rcnfsserver status
Checking for kernel based NFS server:                                  unused
```

Unlike **inetd**, the NFS server cannot run standalone. It also requires the **nfslock** service to be running. Use the **chkconfig** command to query whether these services are configured to run:

```
# chkconfig | grep nfs
nfs                     off
nfslock                 off
nfsserver               off
```

In this example, they are not. They can be turned on with the **chkconfig** command:

```
# chkconfig nfslock on
# chkconfig nfsserver on
```

Now the directories that are to be exported through NFS must be set. This is done in the configuration file /etc/exports. Add a line for each directory you wish to export. The value `*(ro)` means that any workstation with an NFS client can have read-only access to the directory

```
# cd /etc
# vi exports            // add 4 lines
# See the exports(5) manpage for a description of the syntax of this file.
# This file contains a list of all directories that are to be exported to
# other computers via NFS (Network File System).
# This file used by rpc.nfsd and rpc.mountd. See their manpages for details
# on how make changes in this file effective.
/mnt/sles8cd1       *(ro)
/mnt/sles8cd2       *(ro)
/mnt/sles8cd3       *(ro)
/mnt/sles8sp2       *(ro)
```

Now start the NFS server for this session with the **rcnfsserver** command:

```
# rcnfsserver status
Checking for kernel based NFS server:                                unused
# rcnfsserver start
...
                                                                     done
```

You should now be able to mount these directories remotely using NFS.

## Set up an FTP server

Either an NFS or FTP server can be used. Both work well. There are various trade-offs which are negligible. Setting these services up depends on the specifics of your distribution.

In order to set up an FTP server on SLES-8, the following steps are performed. First check that an FTP server is installed with the **rpm -qa** command piped to **grep**:

```
# rpm -qa | grep ftpd
vsftpd-1.1.0-21
```

The FTP server is typically started from the Internet super-server **inetd** (though some systems use **xinetd**, which often has a configuration directory, /etc/xinetd/), so verify that **inetd** is running. If it is not (as is the case with SLES-8), it can be set to start with the **chkconfig** command:

```
# rcinetd status
Checking for inetd:                                                  unused
chkconfig inetd on
```

Now the FTP server must be set to listen in the /etc/inetd.conf file. This is done by uncommenting one line (delete the leading #):

```
# cd /etc
# vi inetd.conf            // uncomment one line
...
# These are standard services.
#
# ftp   stream  tcp     nowait  root    /usr/sbin/tcpd  in.ftpd
ftp     stream  tcp     nowait  root    /usr/sbin/tcpd  vsftpd
...
```

By default, the **vsftpd** daemon only allows anonymous FTP. This must be changed in the /etc/vsftpd.conf file:

```
# vi vsftpd.conf            // uncomment two lines
# Example config file /etc/vsftpd.conf
#
# The default compiled in settings are very paranoid. This sample file
# loosens things up a bit, to make the ftp daemon more usable.
#
# Allow anonymous FTP?
anonymous_enable=YES
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES
...
```

Now start the **inetd** server manually:

```
# rcinetd start
Starting inetd                                                    done
```

Now you should be able to FTP in as a non-root user. If you also want to FTP in as root, delete or comment the root entry in the /etc/ftpusers file.

## Fill in the Linux installation worksheet

You should now have enough information to fill in the worksheet in Table 6-2. This worksheet will be handy when you start the Linux install.

*Table 6-2 Linux installation worksheet*

| Description | Value |
|---|---|
| REXX EXEC to IPL from reader (under z/VM only) | |
| OSA or Guest LAN addresses (usually three) | |
| OSA Express port name (for OSA devices only) | |
| Fully qualified host (DNS) name | |
| Linux IP address | |
| Subnet mask | |
| Gateway IP address | |
| DNS server IP address | |
| MTU size | |
| NFS or FTP server with install CDs | |
| File path to distribution CD | |
| File path to SP2 update CD | |
| FTP server user ID/password (for FTP server only) | |
| Installation method (X Window, VNC or ssh) | |
| TCP/IP address of X server (X Window method only) | |
| DASD addresses to be used | |

You may also want to write down each DASD and corresponding device and filesystem that will be used. When the Linux DASD device driver adds DASD, it takes the addresses and sequentially assigns them to files in the /dev directory starting at dasda, then to dasdb, and so on. The 27th and 28th DASD are assigned to /dev/dasdaa and /dev/dasdab, and so on.

Table 6-3 contains a worksheet that you may find useful to fill in to describe which filesystems the various DASD should be mounted over.

*Table 6-3 DASD and filesystem worksheet*

| DASD address | Linux device file name (for example, /dev/dasda) | Filesystem over which to mount |
|---|---|---|
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

## 6.3 z/VM first steps

**Note:** If you are installing Linux in an LPAR, you can skip this section and go to 6.4, “Prepare and IPL Linux in an LPAR” on page 99.

Once you have a z/VM user ID defined, the first steps on z/VM are to get access to the three IPL files, optionally to create a REXX EXEC to start installation, and to start the installation.

To IPL Linux from the z/VM reader, three files need to be punched, in the following order. The file names on the SuSE SLES-8 CD are specified:

1. The Linux kernel—**vmrdr.ikr**
2. A parameter file—**parmfile**
3. An initial RAMdisk—**initrd**

You will have to move the three files to be punched to the reader over to VM. The kernel and RAMdisk are moved in binary, while the parameter file must be converted from ASCII to EBCDIC. If you do not have access to the **FTP** command on z/VM, it can be accessed with the following commands:

```
link tcpmaint 592 592 rr
acc 592 f
```

Once you have access to FTP from VM, copy the files over in fixed 80-byte records. For example, if your FTP server is 9.117.99.3 and the directory over which SLES-8 CD1 is mounted is /mnt/sles8cd1, the following **FTP** commands will copy the files to z/VM with file names and types that are intuitive:

```
ftp 9.117.99.3
USER (identify yourself to the host): <FTP user name>
Password: <password>
ftp> cd /mnt/sles8cd1/boot
ftp> asc
ftp> locsite fix 80
ftp> get parmfile sles8.parmfile
ftp> bin
```

```
ftp> get initrd sles8.ramdisk
ftp> get vmrdr.ikr sles8.kernel
ftp> quit
```

It is recommended, but not necessary, to create a REXX EXEC that punches these files into the reader and then IPL from it. Following is an example named **SLES8 EXEC**:

```
/* REXX EXEC for loading SUSE SLES-8      */
say 'Loading files for SuSE SLES-8 into reader...'
'CP CLOSE RDR'
'CP PURGE RDR CLASS L'
'CP SPOOL PUN * RDR CLASS L'
'PUNCH SLES8 KERNEL A (NOH'
'PUNCH SLES8 PARMFILE A (NOH'
'PUNCH SLES8 RAMDISK A (NOH'
'CP SPOOL RDR KEEP CLASS L'
'CP IPL 00C CLEAR'
```

If the correct three files are already in the reader and you want to IPL the same installation system, you can bypass the EXEC and simply enter **IPL 00C CLEAR**.

To run the exec, just enter **sle8** from the command line:

```
sles8
Loading files for SuSE SLES-8 into reader...
0000003 FILES PURGED
RDR FILE 0004 SENT FROM LINUX4   PUN WAS 0004 RECS 027K CPY  001 L NOHOLD NOKEEP
RDR FILE 0005 SENT FROM LINUX4   PUN WAS 0005 RECS 0002 CPY  001 L NOHOLD NOKEEP
RDR FILE 0006 SENT FROM LINUX4   PUN WAS 0006 RECS 088K CPY  001 L NOHOLD NOKEEP
0000003 FILES CHANGED
hwc low level driver: can write messages
hwc low level driver: can not read state change notifications
hwc low level driver: can receive signal quiesce
hwc low level driver: can read commands
hwc low level driver: can read priority commands
Linux version 2.4.19-3suse-SMP (root@s390l5) (gcc version 3.2) #1 SMP Thu Oct 17
 11:03:16 UTC 2002
We are running under VM (31 bit mode)
```

Linux should begin booting, as shown above. If there is a problem, be sure you FTPed the files over correctly with the proper attributes. For reference, the CMS files are shown using the **filelist** command:

```
filel sles8 *
Cmd   Filename Filetype Fm Format Lrecl    Records     Blocks
      SLES8    EXEC     A1 V         50         10          1
      SLES8    RAMDISK  A1 F         80      87865       1717
      SLES8    KERNEL   A1 F         80      27060        476
      SLES8    PARMFILE A1 V         50          2          1
```

## 6.4 Prepare and IPL Linux in an LPAR

**Note:** If you are installing Linux under z/VM, you can skip this section and go to 6.5, “Linux installation and customization” on page 105.

It is possible to run Linux for zSeries natively in an LPAR (zSeries or S/390) as a standalone operating system. The install process for depends on the abilities of the Hardware Management Console (HMC) and the media types available.

Linux can be installed:

- By IPLing from tape (see 6.4.4, “Tape preparation” on page 101)
- By IPLing from DASD (see 6.4.5, “DASD preparation” on page 104)
- From CD or the CD code available on FTP (see 6.4.6, “Booting Linux from the HMC” on page 105)

We cover the process of building an IPL-able tape and a tape image on DASD for SuSE Linux SLES 8 for zSeries, IPLing Linux from the install media, and installing the Linux code. The actual installation and customization of Linux is the same, regardless of whether it is installed as a standalone OS in an LPAR or under VM.

When running Linux as a standalone operating system in its own LPAR, Linux is not running under some other operating system such as z/VM or z/OS—rather, it is run as a self-standing operating system on the zSeries or S/390 hardware.

A consideration when running Domino in a Linux LPAR is that the number of LPARs depends on the hardware. Other applications on the machine may reduce the number of LPARs available.

Another consideration is how busy the Domino application within the LPAR is projected to be. An underutilized Linux running in an LPAR can be a waste of resources. Linux running under z/VM can handle dynamic loads more readily and with greater optimization of resources.

### 6.4.1 Assumptions

This section is written for the person installing Linux and does not cover how to set up an LPAR on zSeries hardware. The following assumptions are made:

- An LPAR has already been configured for Linux with the proper resources.
- The person installing Linux has access to the Hardware Management Console (HMC).
- The LPAR has access to a tape drive or a DASD which contains the IPL-able installation image of Linux.
- The installer has access to the z/OS or z/VM operating system and is able to submit JCL and write to tape or DASD.
- The z/OS operating system has FTP access to transfer files from a workstation or z/OS. This can be client access if the workstation is running an FTP server.

### 6.4.2 Skills needed

A variety of skills are needed to install Linux natively on zSeries hardware. These include the following:

- A systems programmer or operations resource capable of creating an LPAR on the zSeries hardware.
- A systems programmer or operations resource with access to the HMC.
- A z/VM or z/OS user with the ability to FTP, run JCL, mount tapes, or create an IPL-able DASD volume. This person must also understand the operating system under which they are working.
  - For VM, this means having a solid working knowledge of CMS and CP commands. This person should also have access to an ID with a high privilege class, allowing system-level operations. Knowledge of REXX is also a plus.
  - The z/OS system programmer should have a solid working understanding of JCL, and have the access necessary to do system-level tasks

- A Linux administrator knowledgeable about the installation and configuration of a Linux system.
- A networking resource with a solid understanding of the local network and the connection options available on the zSeries hardware.

### 6.4.3 FTP installation files to z/OS

The first step in creating either an bootable Linux tape or disk is to acquire the three files needed from the distribution CD. Our CD was mounted on a Linux workstation and the FTP command was issued under TSO (ISPF option 6):

```
ftp <workstation FTP server>
(enter id and password at prompt)
ftp> locsite lrecl=1024 blksize=8192 recfm=fb track pri=260 sec=50
ftp> bin
ftp> get
Remote file cd1/boot/initrd
Local file linux390.initrd
125 Storing data set <hilevqualifier>.LINUX390.INITRD
ftp> locsite lrecl=1024 recfm=f blksize=1024 track pri=1
ftp> get
Remote file get cd1/boot/parmfile
Local file linux390.parmfile
125 Storing data set <hilevqualifier>.LINUX390.PARMFILE
ftp> locsite recfm=fb lrecl=1024 blksize=8192 bl pri=200 sec=100
ftp> get
Remote file cd1/boot/tapeipl.ikr
Local file linux390.tapeipl.ikr
125 Storing data set <hilevqualifier>.LINUX390.TAPEIPL.IKR
ftp> quit
```

Here are the attributes of the files as they were created on the z/OS or OS/390® system:

```
Dataset Name                      Tracks %Used Device Dsorg Recfm Lrecl Blksize
<hilevqualifier>.LINUX390.TAPEIPL    31   100   3390    PS    FB  1024    8192
<hilevqualifier>.LINUX390.INITRD    207   100   3390    PS    FB  1024    8192
<hilevqualifier>.LINUX390.PARMFILE    1   100   3390    PS     F  1024    1024
```

### 6.4.4 Tape preparation

The Linux install tape can be created from either z/VM or z/OS. These files need to have been transferred to the system you are using in order to create the tape:

- The installation RAMdisk file—initrd
- The parameter file—parmfile
- The installation kernel—tapeipl.ikr

#### Creating an IPL tape on z/OS

The `IEBGENER` utility is used to create a tape from the installation files. Before transferring the files to tape, we created a null (empty) file with tape marks at the beginning of the tape. This destroyed any volume label that might have been on the tape.

In a typical z/OS or OS/390 production environment, you will have to bypass tape volume protection in order to write to the tape.

You must be able to specify the “bypass label processing” option (BLP) which was used on DD statement SYSUT2 in the sample job**:**

```
//LINUXTP JOB (999,POK),'CREATE NL',NOTIFY=&SYSUID,
// CLASS=A,MSGCLASS=T,MSGLEVEL=(1,1)
//STEP0 EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DUMMY,DCB=(RECFM=F,BLKSIZE=1024)
//SYSUT2 DD DISP=(NEW,KEEP),LABEL=(1,BLP),DSN=SUSE.BOOT,
// DCB=(RECFM=F,LRECL=1024),VOL=SER=NLTAPE,
// UNIT=B4F
//SYSIN DD DUMMY
//
```

Other volume protection measures in effect by your installed tape management product may require additional DD statement parameters to be specified.

A boot tape can be created by running the GENER job shown:

```
//LINUXTP JOB (999,POK),'CREATE BOOT TAPE',NOTIFY=&SYSUID,
// CLASS=A,MSGCLASS=T,MSGLEVEL=(1,1)
//IMAGE EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DISP=SHR,DSN=<hilevqualifier>.LINUX390.TAPEIPL.IRK
//SYSUT2 DD DISP=(NEW,PASS),LABEL=(1,NL),DSN=DUMMY,
// DCB=(RECFM=F,LRECL=1024),
// UNIT=B4F
//SYSIN DD DUMMY
//PARMLINE EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DISP=SHR,DSN=<hilevqualifier>.LINUX390.PARMFILE
//SYSUT2 DD DISP=(NEW,PASS),LABEL=(2,NL),DSN=DUMMY,
// DCB=(RECFM=F,LRECL=1024),
// VOL=(,RETAIN,,REF=*.IMAGE.SYSUT2),
// UNIT=B4F
//SYSIN DD DUMMY
//INITRD EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DISP=SHR,DSN=<hilevqualifier>.LINUX390.INITRD
//SYSUT2 DD DISP=(NEW,KEEP),LABEL=(3,NL),DSN=DUMMY,
// DCB=(RECFM=F,LRECL=1024),
// VOL=(,RETAIN,,REF=*.IMAGE.SYSUT2),
// UNIT=B4F
//SYSIN DD DUMMY
//
```

**Note:** There may be environments where a non-labelled tape is simply not possible. If you must write the Linux boot files to a standard-labelled tape, you must then issue the IPL command five times. The first four times will fail, but the label will be skipped over, and the fifth IPL should succeed.

## Creating an IPL tape on VM

The following sequence of FTP commands were used to download the required boot files from a CD on a workstation to prepare a tape to be used as the media or boot of Linux from tape. Note that files are reblocked to a record length of 1024 bytes, and that the kernel file is specific to the tape boot method.

Log on to a virtual machine that has the appropriate class privileges to attach tape drives. IPL CMS, if it is not already running.

Access TCPMAINT's 592 disk to make FTP available to download the files tapeipl.ikr, parmfile, and initrd from the SLES8 distribution CD, as shown:

```
link tcpmaint 592 592 rr
acce 592 d
```

The three files that will be downloaded are quite large and will be used only once to create the tape. We recommend placing them on temporary disk (TDISK) using the following commands to define and format the tdisk:

```
def t3380 as 192 cyl 30
format 192 b
DMSFOR603R FORMAT will erase all files on disk B(192). Do you wish to continue?
Enter 1 (YES) or 0 (NO).
1
DMSFOR605R Enter disk label:
temp
Formatting disk B
30 cylinders formatted on B(192)
```

We allocated 30 cylinders of TDISK space, but later found that 15 cylinders of T3380 would have been enough.

FTP the tapeipl.ikr, initrd, and parmfile to the 192 TDISK you just created. We FTPed the files from a workstation running Linux with the SLES8 CD mounted and an FTP server. We used the FTP client in VM. The kernel and initial RAMdisk files must be blocked at 1024:

```
ftp <ftp.server>
USER (identify yourself to the host): <joeuser>
locsite fix 1024
bin
>>>TYPE i
get cd1/boot/initrd SLES8.INITRD.B
8060928 bytes transferred in 90.287 seconds. Transfer rate 89.28 Kbytes/sec.
get cd1/boot/tapeipl.ikr SLES8.TAPEIPL.B
2328576 bytes transferred in 29.617 seconds. Transfer rate 78.62 Kbytes/sec.
asc
>>>TYPE a
get cd1/boot/parmfile SLES8.PARMFILE.B
54 bytes transferred in 0.017 seconds. Transfer rate 3.18 Kbytes/sec.
quit
221 Goodbye.
```

The files we downloaded to the 192 (B) disk were:

```
LVL 0 - B 192      5400 BLKS 3390 R/W       3 FILES 46%  FILE      1 OF      3
SLES8    INITRD   B1                F  1024       7872     1968  8/25/03 13:47
SLES8    TAPEIPL  B1                F  1024       2274      515  8/25/03 13:48
SLES8    PARMFILE B1                V    52          1        1  8/25/03 13:50
```

Attach a tape drive to a virtual machine. The following CP command is an example of attaching the tape device B30, which must be issued from a privileged VM user ID:

```
ATT B30 LINUXC AS 181
```

In this example, the user ID LINUXC now has a virtual device 181, which is the real tape drive B30. Insert a blank non-labelled tape and issue a rewind command to make sure the tape is rewound to the beginning:

```
REW 181
```

Then the following commands will copy the files to tape:

```
FILEDEF IN DISK SLES8 TAPEIPL B
FILEDEF OUT TAP1 (RECFM F LRECL 1024 BLOCK 1024 PERM)
movefile in out
```

```
FILEDEF IN DISK SLES8 PARM B
movefile in out
FILEDEF IN DISK SLES8 INITRD B
movefile in out
```

A REXX EXEC can also be created that will copy the boot files to the tape device 181:

```
/* */
REW 181
'FILEDEF IN DISK SLES8 TAPEIPL B'
'FILEDEF OUT TAP1 (RECFM F LRECL 1024 BLOCK 1024 PERM)'
'movefile in out'
'FILEDEF IN DISK SLES8 PARMFILE B'
'movefile in out'
'FILEDEF IN DISK SLES8 INITRD B'
'movefile in out'
```

The tape can now be either unloaded using the **TAPE RUN** command, or can be rewound for an immediate IPL.

## 6.4.5 DASD preparation

An alternate method of installing Linux is to use the z/OS ICKDSF utility to install a bootstrap loader and the three Linux install files on DASD. Then you IPL from DASD rather than tape to do the initial installation. There is a tool available online that places a DASD Linux bootstrap program and the kernel, parmfile, and RAMdisk files on DASD using ICKDSF. This DASD volume can then be varied offline from the z/OS LPAR and IPLED from the Linux LPAR through the HMC. We found this to be the best method of installing Linux into an LPAR and would recommend it highly.

To install the initial Linux boot code, get the JCL for assembling the ICKDSF Linux bootstrap program and place it and the Linux install files on DASD. It is available from the author, Rob van der Heij, on the Internet at:

http://home.iae.nl/users/rvdheij/linuxipl.html

Or, access it directly at:

http://home.iae.nl/users/rvdheij/linuxipl.asm

Before using the JCL, check for the most recent copy at the Web site. For reference, the JCL can be directly accessed at this Web site:

http://home.iae.nl/users/rvdheij/linuxipl.txt

In addition to this JCL, the following three files must be downloaded using FTP from the SLES-8 distribution CD:

```
initrd - Initializes a ramdisk image
parmfile - The installation parameter file
tapeipl.ikr - Installation kernel image
```

The sample JCL will set up the Linux bootstrap DASD to emulate tape, allowing an IPL from DASD. It will assemble the bootstrap, write a new VTOC on the volume, write the bootstrap records, and copy the three Linux data sets to the volume.

### Run the JCL in dasd.JCL

Modify the LINUXIPL sample JCL to meet your organization standards. Modify the symbolics **&linux** and **&UID** to match the VOLSER of the DASD volume and the high level qualifier for the datasets containing the initrd, parmfile, and tapeipl.ikr files.

The two symbolics that must be set in this sample JCL:

| | |
|---|---|
| **&Linux** | The DASD VOLSER on which you are installing the image |
| **&UID** | The high level qualifier for the location of the three Linux files (initrd,parmfile,taleipl.ikr) |

Vary the address of the Linux boot device online to MVS. Submit the LINUXIPL JCL.

Upon successful completion of the JCL, vary address of Linux boot DASD offline to z/OS.

Although you will IPL from DASD, the device is not usable in Linux after the initial load. We recommend that you run **dasdfmt** against it and use it as your root device for Linux.

### 6.4.6 Booting Linux from the HMC

zSeries servers with the current microcode engineering level (MCEL) come with an integrated CD-ROM drive in the HMC. (The MCEL can be upgraded on your zSeries hardware in some cases.) The HMC is also enabled for FTP, and can be used for the initial load, provided there is network access to an FTP server on which the code is available.

The main advantage to loading the Linux code using one of these methods is that the Linux administrator does not have to depend on the availability of a person with skills in any of the mainframe operating systems; only knowledge of the HMC is needed. The CD-ROM drive on the HMC is used only to stage the kernel image, parameter file, and RAMdisk image. The installation files of your particular Linux distribution must still be “network reachable” from the target LPAR during the installation process. Once the kernel and RAMdisk are loaded, the installation process is the same, regardless of the kernel boot method used.

To start the process of booting Linux from the HMC, select the CPC that contains the Linux (IFL) LPAR. From the Images Work Area on the HMC, select the target LPAR on which Linux will be installed.

To initiate the load process, double-click the icon labeled **Load from CD-ROM or Server**. For a complete description of this process, see 3.4 in *Linux for S/390 and zSeries: Distributions*, SG24-6264, available on the Web at:

http://www.redbooks.ibm.com/abstracts/sg246264.html

## 6.5 Linux installation and customization

We are now finished with our LPAR-specific and VM-specific discussions. From here on, the information applies to both types of installations unless specified.

This section describes SuSE SLES-8 is being installed under z/VM 4.3. At this point you should have Linux IPLed in memory (RAM disk) either from the z/VM reader or in an LPAR.

Installation of Linux on zSeries hardware is quite similar to installation of SuSE SLES-8 on the PC. A major exception is that the zSeries DASD must be manually formatted in the middle of the installation process.

There are many ways to install Linux, and some assumptions are made in the steps that follow. One assumption is that the amount of Domino data will be larger than that which can be stored on a single DASD; therefore, logical volumes are used. Another assumption is that z/VM virtual disks will be used for swap partitions (if you are installing Linux in an LPAR, virtual disks cannot be used, and we discuss that issue a bit later).

Given these assumptions, the following steps are involved in installing and customizing Linux on zSeries:

- Answer the networking questions.
- Begin the graphical installation process.
- Format the DASD from an ssh or telnet session.
- Complete the graphical installation process.
- Reboot the Linux system from disk and finish the basic install.
- Apply the SLES-8 service pack 2 CD.
- Install the sys_epoll RPM.
- Re-IPL with the new kernel.
- Set up the logical volumes.
- Set up the virtual disk swap.
- Turn off unneeded services.
- Make a copy of the root and /opt filesystems - optional (move to after Domino install?)

### 6.5.1 Answer the networking questions

If you are installing under z/VM, these questions will be asked from a 3270 session. If you are installing in an LPAR, these questions will be asked from the HMC. Either way, you will be presented with a choice of network devices.

**Note:** Installation is easier if you have all the relevant information handy. A worksheet is provided in Table 6-2 on page 97 for this purpose. If you have not filled that out yet, this would be a good time to do so.

```
=                                                          =
==-    Welcome to SuSE Linux Enterprise Server 8 for zSeries      -==
=                                                          =

Please select the type of your network device:
0) no network
1) OSA Token Ring
2) OSA Ethernet
3) OSA-Gigabit Ethernet or OSA-Express Fast Ethernet
4) Channel To Channel
5) Escon
6) IUCV
8) Hipersockets
9) Show subchannels and detected devices
Enter your choice (0-9): 3
```

The most common choices are `3` for OSA Express cards (or OSA-2s set up in QDIO mode), `4` for (virtual) channel to channel devices, or `9` for HiperSockets.

If you answer 3 or 9, you will be asked if you want to read the IBM network device driver license and agree with it. You must answer `yes` to both questions or networking will not be installed:

```
To set up the network, you have to read and confirm the license information
of the network device driver provided by IBM.
Do you want to see the license (Yes/No) ? yes
..
Do you agree with this license (Yes/No) ? yes

Ok, now we can set up the network configuration.
...
```

Very often, the install program will detect the correct addresses based on the network type. The following example shows this case; the default value is at the end in parenthesis.

If the default value is correct, you can just press Enter twice (once brings you to the VM READ prompt and the second is taken) and that value will be used.

```
Ok, now we can set up the network configuration.
First OSA Express or Gigabit Ethernet Channels that were detected:
Device Addresses CHPID(s)
            2c08
            2c09
            2c0a
...
Enter the device addresses for the qeth module, e.g. '0x2c08,0x2c09,0x2c0a'
(0x2c08,0x2c09,0x2c0a): <Enter><Enter>
```

You will then be prompted for the port name. On a real OSA card, the first system to specify a port name physically sets that name (up to 8 characters) on the OSA card. After it has been set, any system wanting to share the card must use the same name. If you get the name wrong, you will probably see the error message: `qeth: received an IDX TERMINATE on irq 0x0/0x1 with cause code 0x22 -- try another portname`.

With Guest LANs the port name is not critical, but you must specify something so the name of the Guest LAN is recommended. Following is an example of specifying the correct port name:

```
Please enter the portname(case sensitive) to use(suselin7): OSA2C00
...
qeth: Trying to use card with devnos 0x2C08/0x2C09/0x2C0A
qeth: Device 0x2C08/0x2C09/0x2C0A is an OSD Express card (level: 0330)
with link type Gigabit Eth (portname: OSA2C00)
Module qeth loaded, with warnings
qeth                  153756   0 (unused)
qdio                   33652   1 [qeth]
ipv6                  246300  -1 [qeth]
eth0 detected!
```

Answer all the remaining networking questions (using the installation worksheet on page 97). The broadcast address is not included on the worksheet because it is calculated from the IP address and the subnet mask, therefore, you can just take the default value.

When you are finished, your answers will be summarized as shown in the following example. Check them over and answer: `yes`. (If you have made a mistake, you can answer: `no` and the questions will be asked again.)

```
Configuration for eth0 will be:
Full host name   : linuxa.itso.company.com
IP address       : 184.156.23.73
Net mask         : 255.255.254.0
Broadcast address: 184.156.24.255
Gateway address  : 184.156.23.92
DNS IP address   : 184.156.23.7
DNS search domain: itso.company.com
MTU size         : 1500
Is this correct (Yes/No) ? yes
```

You will be asked for a temporary root password:

```
Please enter the temporary installation password: <secret>
```

You will be asked for the type of installation server. NFS or FTP are recommended. Again you will be given a summary.

In this example, an NFS server is referenced. Answer: `yes` when the information is correct.

```
Please specify the installation Source:

 1) NFS
 2) SAMBA
 3) FTP
 0) Abort

Choice:1
...
Is the following correct?

 Installation Source: nfs
 IP-Address: 9.117.99.3
 Directory: /mnt/sles8cd1

Yes/No: yes
```

You will be asked for the type of terminal. X-Window (**1**) is recommended. You will be asked for the IP address of the desktop. Be sure there is an X server session started on this desktop system. If it is a Linux PC, one will probably be running. If it is a Windows PC, you will need third-party X server software such as Hummingbird® eXceed. If you need the IP address from a Windows machine, use the **ipconfig** command from a DOS prompt.

```
Which terminal do want to use?

 1) X-Window
 2) VNC (VNC-Client or Java enabled Browser)
 3) ssh
 Choice:1

Please enter the IP-Number of the host running the X-Server:184.156.23.151
```

The SuSE SLES-8 installation process now has enough information to begin **yast**.

## 6.5.2 Begin the graphical installation process

> **Note:** A graphical installer uses a great deal of CPU resource and, depending on the network bandwidth between the workstation and the system, it could slow down the install process. This could be a concern if you are installing from a remote location.

Two X Windows should appear on your X server. The one on top should be the SuSE End User License Agreement. Click **Accept**. You will then be prompted for a language from the main installation window. Choose the relevant language (the default of English (US) is what we accepted), and click **Accept** again.

Next, you will then be prompted for the DASD devices to be used when the DASD device driver module is installed as shown in Figure 6-2 on page 110. At a minimum, specify all the DASD assigned to the z/VM user ID and click **Load Module** (but do *not* click Accept yet - refer to 6.5.4, “Complete the graphical installation process” on page 111 for more information). The specified DASD that exists should now be displayed in the lower portion of the window.

You may want to leave some extra “slots” so additional DASD can be added more easily. In the example shown in Figure 6-2 on page 110, we have DASD defined at addresses 200-20f

and 210-213, but we specify **dasd=200-21f** so additional DASD can be added later at addresses 214-21F without having to update the zipl.conf file and to run the **zipl** command.

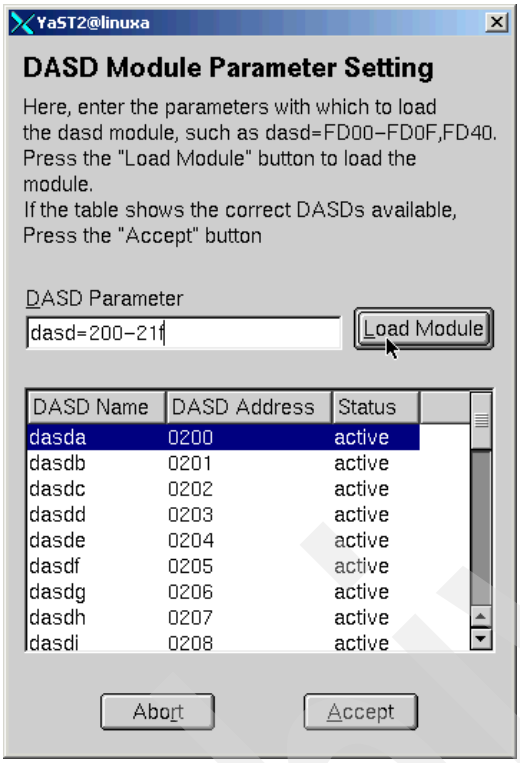



*Figure 6-2 DASD Module Parameter Setting panel*

## 6.5.3 Format the DASD from an ssh or telnet session

The SLES-8 installation process cannot format and partition zSeries DASD. Therefore, you must perform this task *manually* from an ssh or telnet session. In this example, we are installing the root filesystem (/) onto the device /dev/dasda1 and optional software (/opt) onto the device /dev/dasdb1.

Swap space and logical volumes for Domino data will be set up later. Therefore, we want to format the first two DASD (/dev/dasda and /dev/dasdb) and create a single partition out of each (/dev/dasda1 and /dev/dasdb1).

We ssh into our new Linux through PuTTY from a Windows desktop and display the DASD devices with the following command: **cat /proc/dasd/devices**. We then format the DASD with the **dasdfmt** command, using the **-b** flag to set a 4 KB block size and the **-f** flag to specify the device to be formatted. Next, we use the **d** command with the **-a** flag to create a single partition from all available tracks:

```
# cat /proc/dasd/devices
0200(ECKD) at ( 94:  0) is dasda : active at blocksize: 4096, 600840 blocks, 2347 MB
0201(ECKD) at ( 94:  4) is dasdb : active at blocksize: 4096, 600840 blocks, 2347 MB
0202(ECKD) at ( 94:  8) is dasdc : active at blocksize: 4096, 600840 blocks, 2347 MB
...
# dasdfmt -b 4096 -f /dev/dasda
Drive Geometry: 3338 Cylinders * 15 Heads =  50070 Tracks

I am going to format the device /dev/dasda 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
Formatting the device. This may take a while (get yourself a coffee).
Finished formatting the device.
Rereading the partition table... ok
# fdasd -a /dev/dasda
auto-creating one partition for the whole disk...
writing volume label...
writing VTOC...
rereading partition table...
```

The same two steps are done for /dev/dasdb.

Now that the two DASD onto which Linux will be installed are formatted and partitioned, you can exit the ssh or telnet session and return to the graphical installation process: # **exit**

> **Important:** If you are installing Linux in an LPAR and not under z/VM, you may want to use the **fdasd** command without the **-a** flag to interactively create a partition that will be used for a swap space; refer to 6.6.1, “Set up swap space on an LPAR” on page 125 for more information about this topic.

### 6.5.4 Complete the graphical installation process

When you return to the DASD Module Parameter Setting panel shown in Figure 6-2 on page 110, you can now click **Accept**. You will probably see the warning panel shown in Figure 6-3. This warning appears regardless of whether or not Linux was previously installed. Accept the default of **New Installation** and click **OK**.

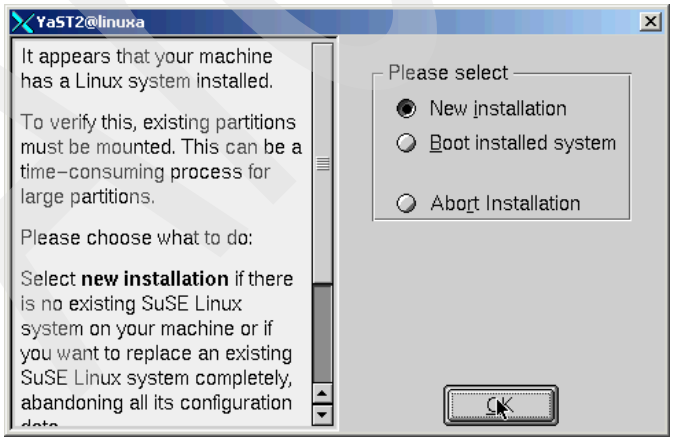



*Figure 6-3 New installation warning panel*

You now see the main Installation Settings window that is shown in Figure 6-4 on page 112. This is the base screen from which partitioning (filesystems), software (packages), and the time zone are customized.

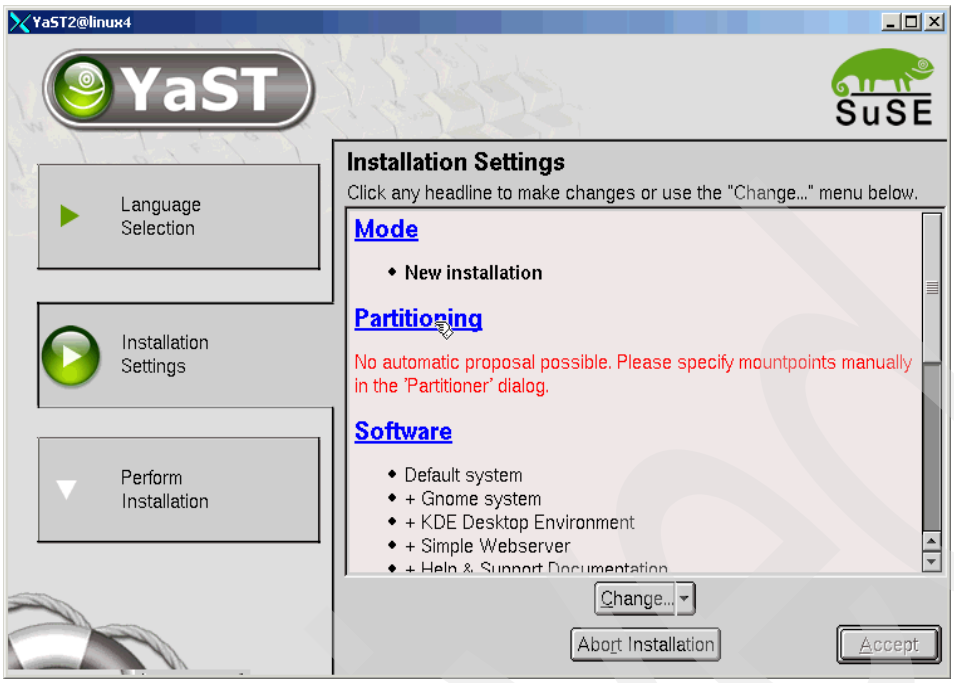



*Figure 6-4 Installation settings panel*

Click **Partitioning**; this brings up the Expert Partitioner window shown in Figure 6-5. You will want to select each DASD partition to be assigned to a filesystem and choose how it will be formatted.

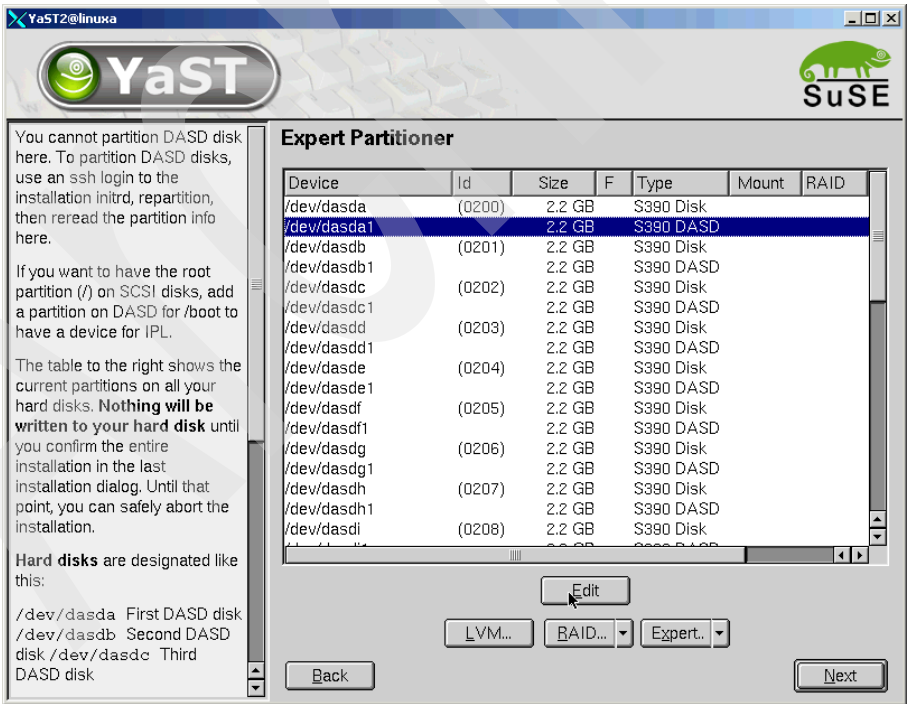



*Figure 6-5 Expert Partitioner panel*

Partition 0 for each DASD represents the disk itself (for example, /dev/dasda). If you used the **fdasd -a** command, as recommended, to carve each disk into a single partition, each disk

should have a corresponding Partition 1 (for example, /dev/dasda1). These are the partitions you will want to work with.

For example, if you select /dev/dasda1 and click **Edit**, you should see the panel shown in Figure 6-6.

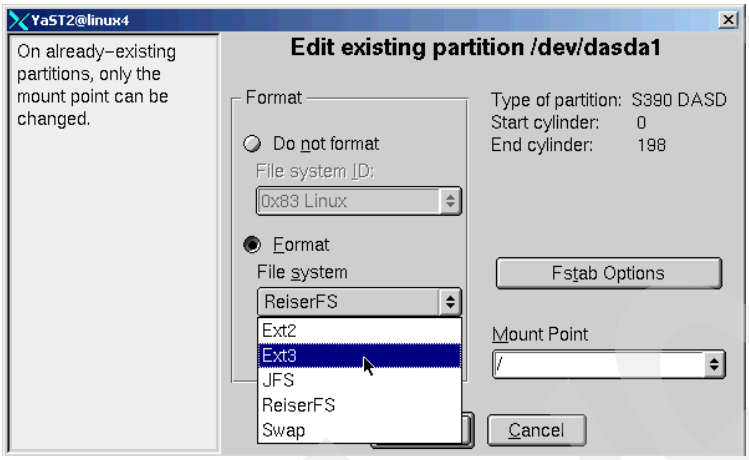



*Figure 6-6 Edit partition panel*

Always choose **Format**, even though you have already formatted the DASD with the **`dasdfmt`** command. This allows you to choose a filesystem type. A type of ext2 or ext3 is recommended, mainly because you can remount any filesystem that is one of these types as the other type. The ext3 filesystem has a journal which allows for a more rapid and reliable recovery from a hard crash, while the ext2 filesystem has the best performance.

In our installation, we choose an ext3 filesystem type for /dev/dasda1 which is mounted over the root filesystem (/) and for /dev/dasdb1 which is mounted over /opt where Domino will be installed. Logical volumes that will be used for Domino data are addressed in 6.5.9, “Set up logical volumes” on page 121, and swap space is addressed in 6.6, “Set up swap space” on page 125.

When you have assigned all DASD to filesystems for the initial installation, click Next. If you have not assigned swap space, you will get a warning. Click No to the question: `Do you want to change this?`. Virtual disk swap space will be added later.

You should now be back at the Installation Settings window, where you should see a summary of your partitioning scheme. Click **Software**, and you should be presented with the window shown in Figure 6-7 on page 114.

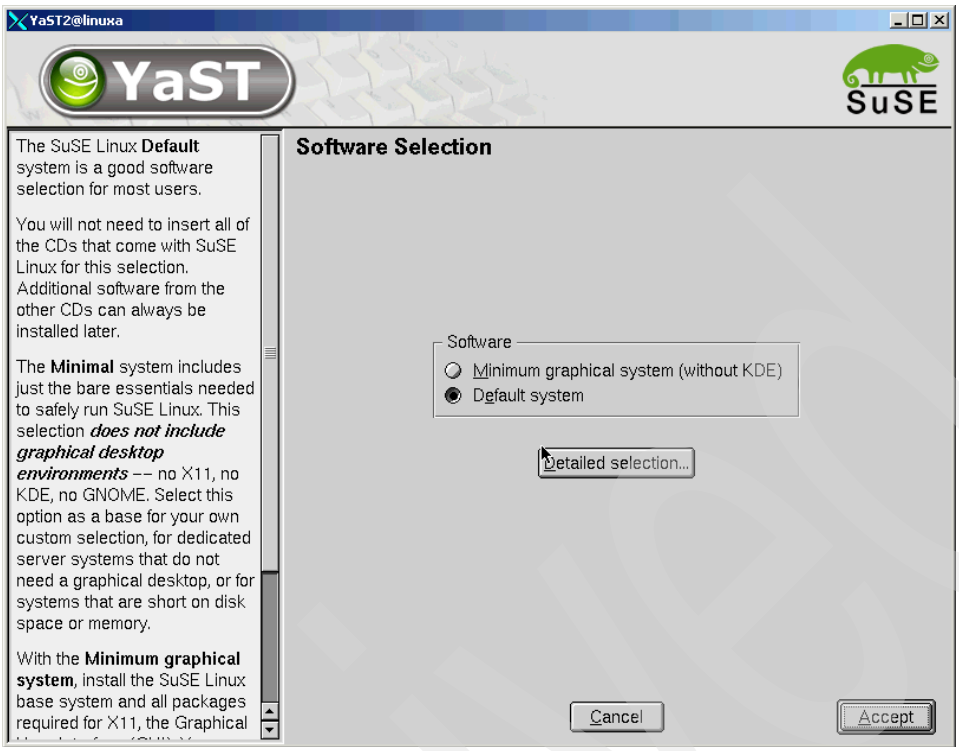



*Figure 6-7 Software Selection window*

You can accept the default system, which seems to be quite adequate for many purposes, or you can customize the packages that will comprise your system. To do this, click **Detailed selection**. This brings up the window in Figure 6-8.

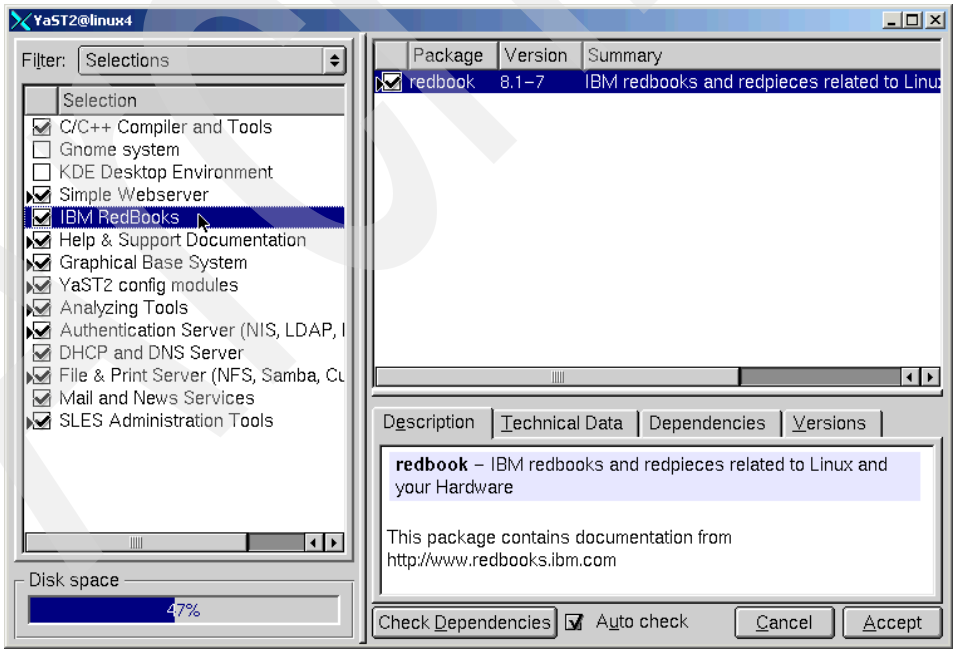



*Figure 6-8 Detailed software selection window*

For our installation, we remove the **KDE Desktop Environment** and **Gnome system**, which are better suited for Linux on other architectures. We add C/C++ Compiler and Tools and, of

course, the **IBM RedBooks** package group! When you are done customizing the packages that will be installed, click **Accept**.

This will again bring you back to the Installation Settings window. The last piece to customize is the time zone. Scroll down and select **Time Zone**. This will bring up the Clock and Time Zone Configuration window. Choose the correct time zone and click **Accept**.

This will again bring you back to the Installation Settings window. Click **Accept** and you should be presented with the warning window shown in Figure 6-9.

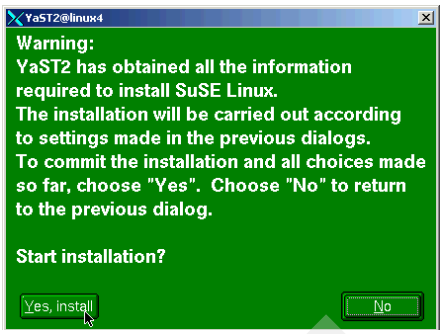



*Figure 6-9 Begin installation warning*

This comes up as a warning because YaST2 will now begin writing to your selected DASD partitions. Click **Yes, install**. The process of loading packages takes some time, usually 20 to 60 minutes, depending on a number of factors.

### 6.5.5 Reboot Linux system from disk and finish basic install

When YaST2 has completed, you will see the message: `Your system will now be shut down. After shutdown reload the system with the load address of your root DASD.`

Click **OK** and return to your 3270 or HMC session. There you will see the message: `Restarting system`. However, it will be the in-memory Linux system IPLed from your reader, so you may very well see the original networking questions awaiting you—do not despair.

On a PC, the device that is booted depends on the boot order. Normally, removing any Linux CD is adequate to ensure the new Linux system that was just copied to disk gets booted. On zSeries hardware, however, the device that is booted (IPLed) is always specified by its address. YaST2 does not decipher this information, so you must *manually* IPL from your boot (/boot) or root (/) filesystem if you did not specify a separate boot filesystem.

In our example, we did not specify a separate boot filesystem, so we IPL the root filesystem (/dev/dasda) which corresponds to minidisk 200. On a z/VM 3270 session, the **#cp** prefix allows CP commands to be sent through the Linux command line to z/VM. Therefore, we IPL our new Linux system with the command:

```
#cp ipl 200
```

If you are installing on an LPAR, from the HMC you would IPL the root DASD address rather than the Linux install tape address.

When Linux boots from DASD, YaST2 should pick up where it left off. It may continue adding packages from CD 2 and CD 3.

When it is finished, you will be asked to enter the real root password. Enter a strong (non-guessable) password in the two fields supplied and click **Next**. Remember this password! You will then be presented with the Add a new user window shown in Figure 6-10.

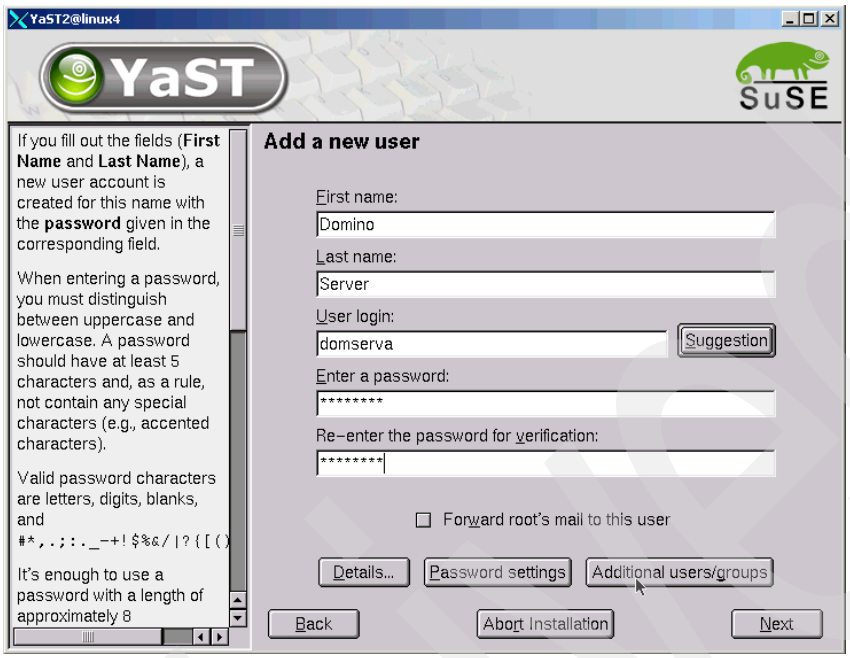



*Figure 6-10 Add a new user window*

It is recommended that you always have a non-root user. This would be a good time to add the non-root user that Domino requires. In this example, we create the user named **`domserva`**. The default group name that Domino uses is the group named **`notes`**.

The Additional users/groups button allows creation of a new group; however, it does not appear to allow it to be the new user's primary group. Therefore, we recommend that you add a new group and set it as the primary group later; this step is documented in 7.3, "Pre-installation steps" on page 135. Click **Next** again.

Now you will see a window entitled Writing the system configuration. The administration command **`SuSEconfig`** is run and you see the output.

You will again see the Installation Settings window with the sections Network interfaces and Printers. You should not have to modify either of these. Select **Next**. You should see a window that says Saving settings that goes away in a few seconds. Your system will reboot again, but this time it knows which DASD to reboot from.

At this time you should be able to start an ssh session to your new Linux image, which is now on DASD.

## 6.5.6 Apply the SLES-8 service pack 2 CD

Prior to June 2003, you would have been done with the SLES-8 installation at this point. However, a service pack 2 (sp2) update CD was subsequently released. This is normally applied with YaST2.

**Important:** If you are accessing an FTP server rather than an NFS server, an extra step is necessary: you must first manually obtain and install the yast2-online-update RPM. Otherwise, you will not be able to get past the FTP credentials screens.

In this example, it is assumed the SP2 CD is mounted over /mnt/sles8sp2cd1:

```
# cd /usr/src
# ftp <your.ftp.server>
ftp> Name : <user name>
ftp> Password: <your password>
ftp> cd /mnt/sles8sp2cd1
ftp> cd s390/update/SuSE-SLES/8/rpm/s390
ftp> mget yast2-online*
mget yast2-online-update-2.6.15-9.s390.rpm [anpqy?]? y
ftp> quit
# rpm -Fvh yast2-online-update-2.6.15-9.s390.rpm
```

Set the DISPLAY environment variable to point to a desktop with an X server running and start **yast2** in the background. For example, if your X server is running on a desktop PC with IP address 184.156.23.151, the following commands will invoke yast2:

```
# export DISPLAY=184.156.23.151:0
# yast2 &
[1] 3108
```

An X window should appear on your X server with the YaST2 Control Center, as shown in Figure 6-11.


*Figure 6-11 YaST2 Control Center*

Click **Patch CD Update** and you should be presented with a new window, as shown in Figure 6-12 on page 118.

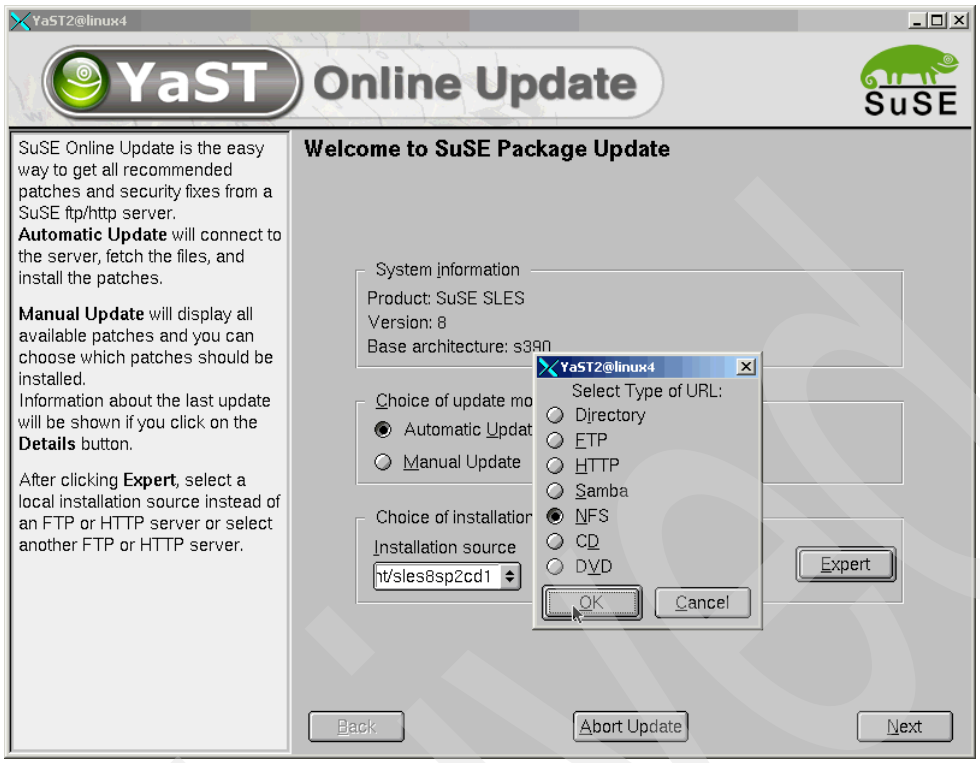



*Figure 6-12 YaST2 Package Update window*

You have a choice of two update modes: manual and automatic. If you choose manual, you must first manually install the yast2-online-update RPM.

We chose to document the automatic mode, so click **Automatic Update** and then **Expert** in the Choice of Installation Source area. You will see the small window shown in the middle of Figure 6-12. Choose the type of URL—probably either FTP or NFS—and click **OK**. When the next window asks for the NFS or FTP credentials, supply them (refer to Table 6-2 on page 97, if you filled it out) and click **Next**.

If you are accessing an FTP server with a user ID other than anonymous, you will have to uncheck the anonymous check box in the middle of the window shown on the left side of Figure 6-13 on page 119.

Enter the user name and password and click **OK**. You will then be presented with another window shown on the right side of the figure. Again enter the user name and password, even though the prompt is asking for “Code” and password.

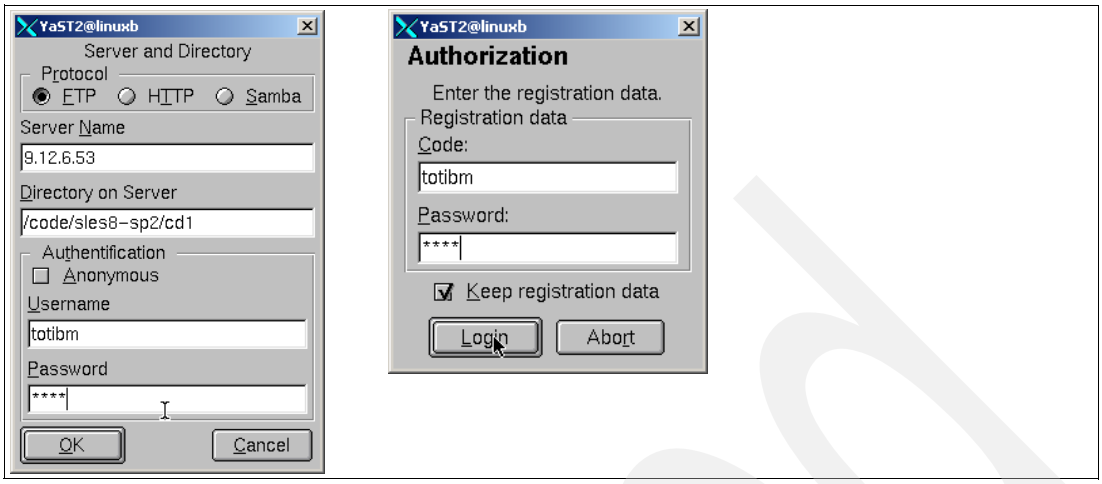



*Figure 6-13 FTP authorization screens*

Only a single patch will be applied: patch-5185. You will see the message: `Please restart the online update to get all available patches`. Click **OK**. You will get another message. Click **OK** again. The **`SuSEconfig`** command will run again and eventually be brought back to the control center shown in Figure 6-11 on page 117.

Again click **Patch CD Update** and go through the same steps again, pointing to the same NFS or FTP server. This time through you should see many patches being applied, as shown in Figure 6-14.

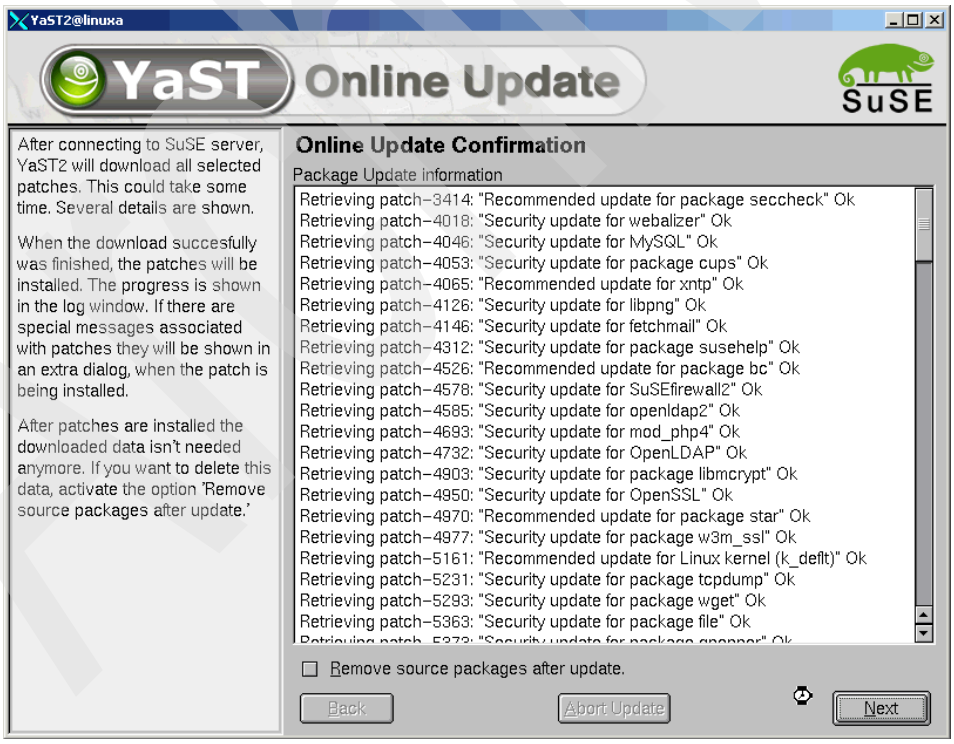



*Figure 6-14 YaST2 Online Update Confirmation window*

You will see a number of warning screens appear. The most relevant is the first, which gives the following message:

```
Attention: Run mkinitrd and zipl after installation!
```

This process can also be lengthy, on the order of 20 to 60 minutes. When the process is complete, you should see the message: `Installation Successful.` Click **OK** to that message and then click **Next**. The **`SuSEconfig`** command will again be run automatically and you will return to the Control Center. Click **Close**.

### 6.5.7 Install the sys_epoll RPM

Return to the ssh session. Even though you just applied all recommended patches to your SLES-8 system, you still have to manually install the sys_epoll RPM (epoll-1.0-9.s390.rpm) that Domino needs. This can be accomplished with yast, but we describe how to do it from the command line. Therefore, you must access the CD either through FTP or NFS from an ssh session.

The following example uses NFS. Verify that nothing is mounted over the directory /mnt and mount the SP2 CD. The RPM can be directly installed from the NFS-mounted directory /mnt/s390/update/SuSE-SLES/8/rpm/s390 using the **`rpm`** command. For example:

```
# ls /mnt
.  ..
# mount 9.117.99.3:/mnt/sles8sp2cd1 /mnt
# cd /mnt/s390/update/SuSE-SLES/8/rpm/s390
# rpm -ivh epoll-1.0-9.s390.rpm
epoll                       ##################################################
```

### 6.5.8 Re-IPL with the new kernel

Run the **`mkinitrd`** and **`zipl`** commands to create a new initial RAMdisk and to update the IPL record:

```
# mkinitrd
using "/dev/dasda1" as root device (mounted on "/" as "ext3")

Found ECKD dasd, adding dasd eckd discipline!

Note: If you want to add ECKD dasd support for later mkinitrd
calls where possibly no ECKD dasd is found, add dasd_eckd_mod
to INITRD_MODULES in /etc/sysconfig/kernel
...
Run zipl now to update the IPL record!

# zipl
building bootmap    : /boot/zipl/bootmap
adding Kernel Image : /boot/kernel/image located at 0x00010000
adding Ramdisk      : /boot/initrd located at 0x00800000
adding Parmline     : /boot/zipl/parmfile located at 0x00001000
Bootloader for ECKD type devices with z/OS compatible layout installed.
Syncing disks....
...done
```

Make a note of the kernel level and build date with the **`uname`** command before you shut down and reboot with the new kernel:

```
# uname -a
Linux linux4 2.4.19-3suse-SMP #1 SMP Wed Nov 6 22:34:43 UTC 2002 s390 unknown
# shutdown -r now
```

After your system reboots, you can get a new ssh session and verify that the kernel has been updated:

```
# uname -a
Linux linux4 2.4.19-4suse-SMP #1 SMP Thu Jun 5 23:01:37 UTC 2003 s390 unknown
```

### 6.5.9 Set up logical volumes

The DASD that will comprise the Domino data logical volumes must first be formatted and partitioned, as with the two DASD onto which Linux was installed. This is done, again by using the **dasdfmt** and **fdasd** commands.

We choose to make one large volume group and create logical volumes from it. Of the 18 packs, two will be used for notesdata, seven for each of mail_1 and mail_2, and the last two for translog. See 4.7, “Placement of other Domino databases” on page 71 for a more detailed discussion.

In order to format multiple DASD simultaneously, we put the dasdfmt jobs in the background. To do this, we found we had to use the /bin/sh shell (shown in the following example with the `sh-2.05b#` prompt). This time, the **-y** flag is added to the **dasdfmt** command, which stacks an answer of “yes”, thus bypassing the question: `Are you sure?`.

In the following example, we format and partition 18 DASD ( /dev/dasdc - /dev/dasdt) using a **for** loop. Note that **i** is a variable that iterates through the elements in the list and is replaced where **$i** is encountered. The loop should iterate quickly. Use the **pstree** command to view the 18 **dasdfmt** processes that are children of the **/bin/sh** shell:

```
# /bin/sh
sh-2.05b# for i in c d e f g h i j k l m n o p q r s t
> do
>   dasdfmt -b 4096 -y -f /dev/dasd$i &
> done
[1] 537
[2] 538
...
[18] 554
sh-2.05b# pstree
init-+-atd
     | ...
     |-sshd---sshd---bash---sh-+-18*[dasdfmt]
     |                         `-pstree
     | ...
...
```

Be sure to wait for the **dasdfmt** jobs to finish (use the **pstree** command again if you are not sure if they are finished). They will finish much faster than if they were done sequentially; however, they will still take approximately 10 to 20 minutes. When they are done, you can **exit** the **/bin/sh** shell.

Now invoke a similar **for** loop. The **fdasd** command does not need to be run in the background, as it runs much faster this way:

```
sh-2.05b# exit
# for i in c d e f g h i j k l m n o p q r s t
> do
>   fdasd -a /dev/dasd$i
> done
...
```

Initialize the new “physical” volume with the **vgscan** command:

```
# vgscan
vgscan -- reading all physical volumes (this may take a while...)
vgscan -- "/etc/lvmtab" and "/etc/lvmtab.d" successfully created
vgscan -- This program does not do a VGDA backup of your volume group
```

Create physical volumes for each DASD with the **pvcreate** command. The shell regular expression /dev/dasd[c-t]1 can be used to address all DASD:

```
# pvcreate /dev/dasd[c-t]1
pvcreate -- physical volume "dasdc1" successfully created
pvcreate -- physical volume "dasdd1" successfully created
...
pvcreate -- physical volume "dasdt1" successfully created
```

Verify physical volumes with the **pvscan** command:

```
# pvscan
pvscan -- reading all physical volumes (this may take a while...)
pvscan -- inactive PV "/dev/dasdc1" is in no VG  [2.29 GB]
pvscan -- inactive PV "/dev/dasdd1" is in no VG  [2.29 GB]
...
pvscan -- inactive PV "/dev/dasdt1" is in no VG  [2.29 GB]
pvscan -- total: 18 [41.25 GB] / in use: 0 [0] / in no VG: 18 [41.25 GB]
```

The volume group named domservb is created with the **vgcreate** command. Using the -s flag, the physical extent size is increased from the default of 4 MB to 16 MB, so that the volume group could be increased to about 1 TB, if necessary.

When the command completes, you can see the volume group has been added to the /dev/ directory. The size of about 41 GB is shown with the **vgdisplay** command:

```
# vgcreate -s 16M domservb /dev/dasd[c-t]1
vgcreate -- INFO: maximum logical volume size is 1023.97 Gigabyte
vgcreate -- doing automatic backup of volume group "domservb"
vgcreate -- volume group "domservb" successfully created and activated
# ls -ld /dev/domservb/
dr-xr-xr-x    2 root     root         4096 Sep  2 14:53 /dev/domservb/
# ls -l /dev/domservb/
crw-r-----    1 root     disk      109,   0 Sep  2 14:53 group
# vgdisplay domservb| grep Size
MAX LV Size           1023.97 GB
VG Size               40.78 GB
PE Size               16 MB
Alloc PE / Size       0 / 0
Free  PE / Size       2610 / 40.78 GB
```

Performance can be greatly increased (in theory) with a striped logical volume. The following recommendations have been made for increasing the performance of logical volumes:

- Spread the host adapters used across all host adapter bays
- Use as many CHPIDs as possible
- Establish connections for each disk to all CHPIDs
- Use at least one host adapter, maximum two per CHPID (more than one host adapter per CHPID requires a director/switch)
- Spread the disks used over all ranks equally
- Avoid (re)using the same resources (CHPID, host adapter, rank) as much as possible

For more details on these performance recommendations, see:

http://www-124.ibm.com/developerworks/opensource/linux390/perf_hints_tips.shtml#subcategory2_1

However, striped logical volumes cannot be extended with the **lvextend** command, so be sure to make each volume large enough up front.

We create logical volumes that are striped across all 18 DASD in the volume group. We express the size of the logical volumes in extents rather than in bytes.

```
# lvcreate --extents 290 --stripes 18 -n domservb /dev/domservb
lvcreate -- INFO: using default stripe size 16 KB
lvcreate -- rounding to stripe boundary size
lvcreate -- doing automatic backup of "domservb"
lvcreate -- logical volume "/dev/domservb/domservb" successfully created
# lvcreate --extents 1015 --stripes 18 -n mail_1 /dev/domservb
lvcreate -- INFO: using default stripe size 16 KB
lvcreate -- rounding to stripe boundary size
lvcreate -- doing automatic backup of "domservb"
lvcreate -- logical volume "/dev/domservb/mail_1" successfully created
# lvcreate --extents 1015 --stripes 18 -n mail_2 /dev/domservb
lvcreate -- INFO: using default stripe size 16 KB
lvcreate -- rounding to stripe boundary size
lvcreate -- doing automatic backup of "domservb"
lvcreate -- logical volume "/dev/domservb/mail_2" successfully created
# lvcreate --extents 252 --stripes 18 -n translog /dev/domservb
lvcreate -- INFO: using default stripe size 16 KB
lvcreate -- doing automatic backup of "domservb"
lvcreate -- logical volume "/dev/domservb/translog" successfully created
```

**Note:** Parameters and other considerations must be taken into account when striping, to be sure that you are getting optimum performance. We do not necessarily recommend striping all of the volumes.

The **lvdisplay** command shows you the details of logical volumes. For example, to see the details of mail_1:

```
# lvdisplay /dev/domservb/mail_1
--- Logical volume ---
LV Name                /dev/domservb/mail_1
VG Name                domservb
LV Write Access        read/write
LV Status              available
LV #                   2
# open                 0
LV Size                16.03 GB
Current LE             1026
Allocated LE           1026
Stripes                18
Stripe size (KByte)    16
Allocation             next free
Read ahead sectors     1024
Block device           58:1
```

The **mke2fs** command makes an ext2 filesystem on each of the four logical volumes. We choose ext2 for performance knowing that recovery time will be longer than with a journaled filesystem such as ext3. If you want to choose a journaled filesystem for faster recovery time, you would include the -j flag in the mke2fs command.

```
# mke2fs /dev/domservb/domservb
mke2fs 1.28 (31-Aug-2002)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
```

```
627744 inodes, 1253376 blocks
62668 blocks (5.00%) reserved for the super user
First data block=0
39 block groups
32768 blocks per group, 32768 fragments per group
16096 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736

Writing inode tables: done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 32 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
# mke2fs /dev/domservb/mail_1
...
# mke2fs /dev/domservb/mail_2
...
# mke2fs /dev/domservb/translog
...
```

If we had not included the DASD addresses 202-213 when we installed, we would have had to update the /etc/zipl.conf file and run **zipl** (remember, we used DASD addresses 200-21F when installing with yast, which still leaves open “slots” 214-21F if more DASD is needed later).

Because all the addresses exist, all the logical volumes will be picked up when Linux is re-IPLed. We verify by inspecting the first nine lines of the zipl.conf file, which contain the parameters for the default IPL:

```
# head -9 /etc/zipl.conf
# Generated by YaST2
[defaultboot]
default=ipl

[ipl]
target=/boot/zipl
image=/boot/kernel/image
ramdisk=/boot/initrd
parameters="dasd=200-21f root=/dev/dasda1"
```

We now need to add the logical volumes to the filesystem table in the /etc/fstab file. It is good to first modify the /etc/fstab file, and to specify only the target directory to the **mount** command in order to test your changes without an IPL. Making a mistake in the /etc/fstab file can easily result in a system that will not IPL, so this method of updating the fstab file is more reliable.

First create the empty directory that the logical volume domservb will be mounted over. Then back up and modify the /etc/fstab file (if you used an ext3 filesystem, the filesystem type in the third field would be ext3 rather than ext2):

```
# mkdir /domservb
# cd /etc
# cp fstab fstab.orig
# vi /etc/fstab          # add 4 lines
/dev/dasda1          /                    ext3       defaults              1 1
/dev/dasdb1          /opt                 ext3       defaults              1 2
/dev/domservb/domservb /domservb          ext2       defaults              1 3
/dev/domservb/mail_1   /domservb/notesdata/mail_1 ext2 defaults            1 3
/dev/domservb/mail_2   /domservb/notesdata/mail_2 ext2 defaults            1 3
/dev/domservb/translog /domservb/notesdata/translog ext2 defaults          1 3
```

```
devpts              /dev/pts            devpts     mode=0620,gid=5       0 0
proc                /proc               proc       defaults              0 0
```

Now test the mount of the /domservb/ directory. Once it is mounted, **cd** into it and create the additional directories that will serve as mount points:

```
# mount /domservb/
# cd /domservb/
# mkdir notesdata
# cd notesdata/
# mkdir mail_1 mail_2 translog
```

Again the mount command is used with one parameter to test the other three entries in the /etc/fstab file. When it succeeds, issuing the mount command with no parameters shows that the four logical volumes are mounted:

```
# mount mail_1
# mount mail_2
# mount translog
# mount
/dev/dasda1 on / type ext3 (rw)
proc on /proc type proc (rw)
devpts on /dev/pts type devpts (rw,mode=0620,gid=5)
/dev/dasdb1 on /opt type ext3 (rw)
shmfs on /dev/shm type shm (rw)
/dev/domservb/domservb on /domservb type ext2 (rw)
/dev/domservb/mail_1 on /domservb/notesdata/mail_1 type ext2 (rw)
/dev/domservb/mail_2 on /domservb/notesdata/mail_2 type ext2 (rw)
/dev/domservb/translog on /domservb/notesdata/translog type ext2 (rw)
```

Test a reboot with the **shutdown** command. The system should come back up fine. You can monitor progress from the z/VM 3270 console.

```
# shutdown -r now
# exit
```

More details on LVM can be found on the SuSE Web site in the paper *LVM - Logical Volume Manager* at:

http://www.suse.de/us/whitepapers/lvm/

## 6.6 Set up swap space

Swap space is often set up differently, depending on whether you are installing Linux on an LPAR or under z/VM. You can query your swap space by looking at the swaps file in the /proc virtual filesystem. For example, the following command shows that there is no swap space active:

```
# cat /proc/swaps
Filename                        Type            Size    Used    Priority
```

### 6.6.1 Set up swap space on an LPAR

On an LPAR, you do not have the luxury of z/VM minidisks nor virtual disks. With the Linux **fdasd** command, you can create a partition of just about any size for swap space. It is easiest to add this with yast during installation. However, you can always add swap space after the install is done.

## 6.6.2 Set up virtual disk swap space under z/VM

A conventional swap *file* can be set up on Linux under z/VM. However, virtual disk allows swap spaces to be part of expanded storage (memory) that can also be paged in and out by z/VM. A small Linux machine size combined with multiple smaller virtual disk swap spaces and adequate paging space seem to be accepted as the best combination that allow you to utilize your zSeries resources most efficiently.

Setting up virtual disk is perhaps trickier than setting up conventional swap space. The tasks can be divided as follows:

- Make virtual disk space available to the z/VM system
- Get virtual disk space on each of the z/VM user IDs
- On Linux, set up the virtual disk as swap spaces

**Tip:** One expert recommends allocating the virtual disks with varying sizes (each 1.5 or 2 times as large as the next smaller disk), and using a higher priority for the smaller disks. However, we did not learn this until after the project completed, so we cannot verify how well it works.

### Make virtual disk space available to the z/VM system

On the z/VM system as a whole, virtual disk space needs to be available. Then each of the Linux user IDs can get some of that space for Linux to work with.

Log on as MAINT and query the current virtual disk limits. The **query vdisk syslim** command may show that the system can create an unlimited amount of virtual disk. For example, the following command shows this is true; 1500000 blocks of virtual disk is in use.

```
query vdisk syslim
VDISK SYSTEM LIMIT IS INFINITE,          1500000 BLK IN USE
```

If you allow the creation of virtual disks by a user, then the following command queries the amount of virtual disk space available to the user.

```
query vdisk userlim
VDISK USER   LIMIT IS INFINITE
```

The commands **SET VDISK SYSLIM INFINITE** and **SET VDISK USERLIM INFINITE** will make these settings. However they will only be in effect for the current system IPL. Figure 6-15 shows the syntax for the **SET VDISK** command.

```
>>--Set--.-VDISK-.--.-Syslim--.--.-Default----------------.----------------><
        '-VDSK--'  '-Userlim-'  |-Infinite---------------|
                                |             .-Blocks-. |
                                |-nnnnnnnnnn--+--------+-|
                                |             '-Blks---' |
                                |-nnnnnnnM----------------|
                                '-nnnnG------------------'
```

*Figure 6-15 CP Set VDISK command syntax*

To make these changes permanent, the command should be added to the bottom of the SYSTEM CONFIG file. Refer to 3.3.8, “Update the SYSTEM CONFIG file” on page 49 for a discussion on how it is updated.

The changes necessary to the file are in the **Vdisk** section of the **Features** Statement:

```
/*********************************************************************/
/*                        Features Statement                         */
/*********************************************************************/

  Features  ,
   Disable ,                        /* Disable the following features */
     Auto_Warm_IPL ,                /* Prompt at IPL always           */
   Retrieve ,                       /* Retrieve options               */
     Default  10,                   /* Default.... default is 7       */
     Maximum  20,                   /* Maximum.... default is 7       */
   MaxUsers noLimit ,               /* No limit on number of users    */
   Enable  ,
     Clear_Tdisk   ,
     Set_Privclass ,                /*    allow SET PRIVCLASS command */
     LogMsg_From_File ,             /*    LOGMSG from SYSTEM LOGMSG   */
   Passwords_on_Cmds  ,
     Autolog  no   ,
     Link     no   ,
     Logon    no   ,
   Vdisk ,
       Syslim infinite    ,
       Userlim infinite
```

## Get virtual disk space on each of the z/VM user IDs

On the Linux user ID side, there are at least two ways that virtual disk swap space can be allocated:

1. Manually
2. By using the SWAPGEN EXEC (written by Dave Jones, Adam Thornton of Sine Nomine Associates, and Phil Smith III of Linuxcare)

Background about the manual method is provided in Appendix E, “Creating virtual disks manually” on page 393.

The SWAPGEN EXEC is run on CMS for each virtual disk swap space that is to be used before Linux is IPLed. This EXEC creates the virtual disk, formats, reserves, and performs the equivalent of the Linux **mkswap** command, all in one VM CMS EXEC. This method is described here, since it appears both cleaner and better accepted by the z/VM and Linux on zSeries communities.

The SWAPGEN EXEC is available on the Internet:

http://sinenomine.net/node/265

From CMS, before Linux is IPLed, use the SWAPGEN EXEC specifying the diagnose driver to create 10 small swap virtual disks. This can be put into your PROFILE EXEC.

Here is a sample PROFILE EXEC that defines 10 virtual disks and then queries the user that is not disconnected whether it wants to IPL Linux:

```
/* Sample PROFILE EXEC for Linux */
 'CP SET RUN ON'
 'SWAPGEN 300 100000 (DIAG'
 'SWAPGEN 301 100000 (DIAG'
 'SWAPGEN 302 100000 (DIAG'
 'SWAPGEN 303 100000 (DIAG'
 'SWAPGEN 304 100000 (DIAG'
 'SWAPGEN 305 100000 (DIAG'
```

```
'SWAPGEN 306 100000 (DIAG'
'SWAPGEN 307 100000 (DIAG'
'SWAPGEN 308 100000 (DIAG'
'SWAPGEN 309 100000 (DIAG'
'PIPE CP QUERY' userid() '| var user'
parse value user with . . dsc .
if dsc = 'DSC' then
  startLinux = 1
else
 do
  say 'Do you want to IPL Linux? Y/N'
  parse upper pull answer .
  if answer = 'Y' then
    startLinux = 1
 end
if startLinux = 1 then
  'CP IPL 200'
else
 do
  'CP SET PF11 RETRIEVE FORWARD'
  'CP SET PF12 RETRIEVE'
 end
```

## On Linux, set up the virtual disk as swap spaces

After you have installed Linux and run the SWAPGEN EXEC for each of the swap spaces, you must configure Linux to use them. This can be done two ways: with the **dasd** script that is provided in "The DASD script" on page 381 (not necessarily recommended for production), or by changing the boot parameters in the /etc/zipl.conf file.

### *Using the dasd script*

The **dasd** script performs different operations. The feature we use is the **add** command. This adds the list of DASD to the available filesystems. Use the **dasd** script with the **add** parameter to add the virtual disks at addresses 300-309, and then with the **list** parameter to examine the new DASD:

```
# dasd add 300-309
# dasd list
0200(ECKD) at ( 94:  0) is dasda : active at blocksize: 4096, 600840 blocks, 2347 MB
...
021e(none) at ( 94:120) is dasdae : unknown
021f(none) at ( 94:124) is dasdaf : unknown
0300(none) at ( 94:128) is dasdag : unknown
0301(none) at ( 94:132) is dasdah : unknown
0302(none) at ( 94:136) is dasdai : unknown
0303(none) at ( 94:140) is dasdaj : unknown
0304(none) at ( 94:144) is dasdak : unknown
0305(none) at ( 94:148) is dasdal : unknown
0306(none) at ( 94:152) is dasdam : unknown
0307(none) at ( 94:156) is dasdan : unknown
0308(none) at ( 94:160) is dasdao : unknown
0309(none) at ( 94:164) is dasdap : unknown
```

Notice that we have used more DASD (42) than there are letters in the alphabet. The DASD driver names the twenty-sixth DASD /dev/dasdz, and when it gets to the twenty-seventh, it begins at /dev/dasdaa. However, the SLES-8 distribution only creates nodes in the /dev/ directory for twenty-six DASD. You can see above that there are no nodes.

```
# ls -l /dev/dasda*
brw-rw----    1 root     disk      94,   0 Nov  5  2002 /dev/dasda
```

```
brw-rw----    1 root     disk      94,   1 Nov  5  2002 /dev/dasda1
brw-rw----    1 root     disk      94,   2 Nov  5  2002 /dev/dasda2
brw-rw----    1 root     disk      94,   3 Nov  5  2002 /dev/dasda3
```

We write a quick script named **mknodDASDaa** to invoke the **mknod** command with major number 94 (DASD) and the correct minor number and file name. This adds nodes for 26 more DASD.

```
# cat mknodDASDaa
#!/bin/sh
# mknodDASDaa - add 26 nodes in the /dev/ directory for additional DASD/Vdisk
let minor=104
for i in a b c d e f g h i j k l m n o p q r s t u v w y z
 do
  for j in ' ' 1 2 3
   do
     mknod /dev/dasda$i$j b 94 $minor
     let minor=minor+1
   done
 done
chmod 660 /dev/dasda*
chgrp disk /dev/dasda*
```

If you need more DASD nodes, you can either type in the above or copy the script (remember **chmod +x**) and run it:

```
# ./mknodDASDaa
```

Examine the /dev/ directory and verify that the new nodes have been created:

```
# ls -l /dev/dasda*
brw-rw----    1 root     disk      94,   0 Nov  5  2002 /dev/dasda
brw-rw----    1 root     disk      94,   1 Nov  5  2002 /dev/dasda1
brw-rw----    1 root     disk      94,   2 Nov  5  2002 /dev/dasda2
brw-rw----    1 root     disk      94,   3 Nov  5  2002 /dev/dasda3
brw-rw----    1 root     disk      94, 104 Sep  4 16:04 /dev/dasdaa
brw-rw----    1 root     disk      94, 105 Sep  4 16:04 /dev/dasdaa1
brw-rw----    1 root     disk      94, 106 Sep  4 16:04 /dev/dasdaa2
brw-rw----    1 root     disk      94, 107 Sep  4 16:04 /dev/dasdaa3
...
brw-rw----    1 root     disk      94, 203 Sep  4 16:04 /dev/dasdaz3
```

You could add the swap spaces interactively using the **mkswap** and **swapon** commands. However, you will want these swap spaces to be established at boot time. Therefore, it is recommended that you back up and modify the /etc/fstab file:

```
# cd /etc
# cp fstab fstab.0904
# vi fstab                    // add the following 10 swap entries
/dev/dasda1          /                    ext3       defaults              1 1
/dev/dasdb1          /opt                 ext3       defaults              1 2
/dev/domservb/domservb /domservb          ext2       defaults              1 3
/dev/domservb/mail_1   /domservb/notesdata/mail_1 ext2 defaults            1 3
/dev/domservb/mail_2   /domservb/notesdata/mail_2 ext2 defaults            1 3
/dev/domservb/translog /domservb/notesdata/translog ext2 defaults          1 3
/dev/dasdag1         swap                 swap       defaults              0 0
/dev/dasdah1         swap                 swap       defaults              0 0
/dev/dasdai1         swap                 swap       defaults              0 0
/dev/dasdaj1         swap                 swap       defaults              0 0
/dev/dasdak1         swap                 swap       defaults              0 0
/dev/dasdal1         swap                 swap       defaults              0 0
/dev/dasdam1         swap                 swap       defaults              0 0
/dev/dasdan1         swap                 swap       defaults              0 0
```

```
/dev/dasdao1          swap                 swap       defaults              0 0
/dev/dasdap1          swap                 swap       defaults              0 0
devpts                /dev/pts             devpts     mode=0620,gid=5       0 0
proc                  /proc                proc       defaults              0 0
```

Before you can utilize the new swap spaces, you must load the diagnose driver with the **modprobe** command. You can then verify that it has been loaded with the **lsmod** command:

```
# modprobe dasd_diag_mod
dasd(diag): DIAG discipline initializing
Sep  5 11:00:47 linuxa kernel: dasd(diag): DIAG discipline initializing
# lsmod
lsmod
Module                  Size  Used by    Tainted: P
dasd_diag_mod           5768   0  (unused)
qeth                  154180   1
qdio                   33652   1  [qeth]
ipv6                  330012  -1  [qeth]
reiserfs              247952   1  (autoclean)
lvm-mod                70724   9  (autoclean)
dasd_eckd_mod          57812  20
dasd_mod               49196  23  [dasd_diag_mod dasd_eckd_mod]
ext3                   92176   1
jbd                    52916   1  [ext3]
linuxa:~ #
```

You can test the changes in the /etc/fstab file for swap spaces via the **swapon -a** command. If all has gone well, your swap spaces should be active. Verify this by examining at the /proc/swaps file:

```
# swapon -a
# cat /proc/swaps
Filename                        Type            Size    Used    Priority
/dev/dasdag1                    partition       49572   0       -1
/dev/dasdah1                    partition       49572   0       -2
/dev/dasdai1                    partition       49572   0       -3
/dev/dasdaj1                    partition       49572   0       -4
/dev/dasdak1                    partition       49572   0       -5
/dev/dasdal1                    partition       49572   0       -6
/dev/dasdam1                    partition       49572   0       -7
/dev/dasdan1                    partition       49572   0       -8
/dev/dasdao1                    partition       49572   0       -9
/dev/dasdap1                    partition       49572   0       -10
```

## Making the swap spaces permanent

Some of the steps you have performed so far will work for the current Linux session, but will not work across IPLs. To ensure that the swap spaces will be the same after a re-IPL (reboot), you must set the dasd diagnose driver to load, and you must set the swap device addresses to be loaded. These tasks are done with the **mkinitrd** and **zipl** commands.

A good way to ensure that the DASD diagnose driver will be loaded is to add it to the file /etc/sysconfig/kernel in the variable INITRD_MODULES, then run **mkinitrd**:

```
# cd /etc/sysconfig/
# cp kernel kernel.orig
# vi kernel -> add
INITRD_MODULES="jbd ext3 dasd_diag_mod"
# mkinitrd
...
- insmod jbd                 (kernel/fs/jbd/jbd.o)
 - insmod ext3                (kernel/fs/ext3/ext3.o)
```

```
 - insmod dasd_mod dasd=$dasd (kernel/drivers/s390/block/dasd_mod.o)
 - insmod dasd_diag_mod       (kernel/drivers/s390/block/dasd_diag_mod.o)
 - insmod dasd_eckd_mod       (kernel/drivers/s390/block/dasd_eckd_mod.o)
Run zipl now to update the IPL record!
```

You will need to edit the file /etc/zipl.conf and run **zipl** to add the VDISK DASD. First back up the /etc/zipl.conf file and add devices 300-309 to the parameter line. Then run **zipl**:

```
# cd /etc
# cp zipl.conf zipl.conf.0904
# vi zipl.conf                  //add devices 300-309
# Generated by YaST2
[defaultboot]
default=ipl

[ipl]
target=/boot/zipl
image=/boot/kernel/image
ramdisk=/boot/initrd
parameters="dasd=200-21f,300-309 root=/dev/dasda1"
...
# zipl
building bootmap    : /boot/zipl/bootmap
adding Kernel Image : /boot/kernel/image located at 0x00010000
adding Ramdisk      : /boot/initrd located at 0x00800000
adding Parmline     : /boot/zipl/parmfile located at 0x00001000
Bootloader for ECKD type devices with z/OS compatible layout installed.
Syncing disks....
...done
```

It is now time to reboot and test your changes:

```
# shutdown -r now
# exit
```

When the system comes back, ssh in and verify the swap spaces are valid with the **cat /proc/swaps** command.

### 6.6.3 Turning off unneeded services

Some services on SLES-8 are running, but are probably not needed. We turned off the following six services with the **chkconfig** command for the duration of the project and the Domino server performed fine:

```
# chkconfig atd off
# chkconfig fbset off
# chkconfig hotplug off
# chkconfig postfix off
# chkconfig smbfs off
# chkconfig xdm off
```

Congratulations! At this point you should be ready to install Domino.

### 6.6.4 Back up your system (optional)

You may consider backing up your two system DASD at this point (200 and 201, in this example). If you were to back them up to two other DASD and your system were to become corrupted somehow, you would be able to restore your system back to its original state. You may also prefer to wait and do this step after installing Domino.

A simple REXX EXEC named **copydisk** can be used that invokes DDR, the DASD dump and restore program. To use the EXEC, simply supply the DASD to be copied from and to.

For example, to copy the DASD at addresses 200 and 201 to 400 and 401, the following two **copydisk** commands would be used:

```
copydisk 200 400
Copy minidisk 200 to 400?
y
z/VM DASD DUMP/RESTORE PROGRAM
HCPDDR711D VOLID READ IS 0X0200
HCPDDR711D VOLID READ IS 0X0400
COPYING   0X0200
COPYING DATA  09/09/03 AT 12.02.09  GMT FROM 0X0200 TO 0X0400
INPUT CYLINDER EXTENTS      OUTPUT CYLINDER EXTENTS
      START       STOP         START        STOP
   00000000   00003337      00000000    00003337
END OF COPY
END OF JOB
copydisk 201 401
...
```

For a listing of the EXEC, see “The COPYDISK EXEC” on page 381.

7

# Domino installation

In this chapter, we describe how to install and set up a Domino server. We cover both running the Domino executable code installation from the Linux console, and setting up your Domino server to either be a new server or an additional server in your existing domain. We also discuss how to install multiple levels of Domino server code on the same system.

We cover the following topics:

- Pre-installation steps
- Installing the executable server code
- Setting up the Domino server
- Starting the Domino server

## 7.1 Receiving the Domino CD

IBM Lotus Domino for Linux on zSeries is packaged on CD media in order to be compatible with all other platforms. Table 7-1 lists some useful material that comes on the product CD.

*Table 7-1 Product information about Domino CD*

| Name | Description |
|---|---|
| start.txt | Where to find online documentation and instructions to install Acrobat |
| readme.txt | Last-minute notes not contained in Release Notes |
| readmes.nsf | Release Notes - Notes database |
| readmes.pdf | Release Notes - Acrobat file |
| Help65_Admin.nsf | Administrator Help - Notes database |
| Install.PDF | Install Guide - Acrobat file |

## 7.2 Checklist

Fill out the list in Table 7-2 before and during your Domino installation. These items are used often, so having them written down one place will make your installation task easier. Make a copy of this page as a worksheet, and feel free to add more items and values if your environment requires them. A more detailed description of our environment can be found in Appendix A, “Our test system” on page 375.

*Table 7-2 Domino installation checklist*

| Item | Our values | Your values |
|---|---|---|
| Domino user account on Linux | domserv# | |
| Group for Domino user on Linux | notes | |
| IP address of your Linux server | 192.192.192.192 (sample) | |
| Path for Notesdata directory | /domserv#/notesdata | |
| Mountpoint for notesdata directory | /domserv# | |
| Path for Transactional log | /domserv#/notesdata/translog | |
| Mount point for transactional log | /domserv#/notesdata/translog | |
| Path for mail<br>or application directory | /domserv#/notesdata/mail<br>or /domserv#/notesdata/appl | |
| Mount point for mail<br>or application directory | /domserv#/notesdata/mail<br>or /domserv#/notesdata/appl | |
| Domino server name | DomServ# | |
| Domino domain name | ITSO | |
| DNS entry for Linux server | Linux#.itso.com | |
| DNS entry for Domino server | domserv#.itso.com | |
| Web server configuration | | |

| | | |
|---|---|---|
| Information about other applications your Domino server needs like:<br>Virus Scanner, C/C++ Toolkit, FAX software, TDP, external Databases (DB2) ... | | |

## 7.3 Pre-installation steps

Be sure to read and refer to the following documentation.

- The release notes that come with your Domino distribution CD have the latest information about the product. Information about Domino for Linux on zSeries is scattered throughout.
- Read the install guide, also on the CD, to become familiar with the entire install process.
- Read Chapter 2, “Planning” on page 13. Be sure that you have the Domino clients on your desktop, as listed in 2.5.2, “Software” on page 19.

If you followed the Linux installation instructions in 6.5.4, “Complete the graphical installation process” on page 111, then you should have a non-root user (for example, domserva) defined to run the Domino server, but not a group (for example, notes). The Domino server will not start with a UID of 0. If you have not created the Domino server user ID, you can use the **useradd** and **passwd** commands to do so.

You can verify the user ID with the **id** command. For example:

```
# id domservb
uid=500(domservb) gid=100(users) groups=100(users)
```

Create the group called notes with the **groupadd** command and make it the Domino user’s primary group using the **usermod -g** command. For example:

```
# groupadd notes
# usermod -g notes domservb
```

Verify the changes by using the **id** command:

```
# id domservb
uid=500(domservb) gid=500(notes) groups=500(notes)
```

We recommend that you have notesdata as the home directory for your Domino user account. If you log on with the Domino user ID, you are already in the notesdata directory when you start and configure (that is, by editing notes.ini) If this is not the case, you can use the **usermod** command to change it:

```
# usermod -d /domservb/notesdata
```

Set it to be owned by the correct group.

```
# ls -ld /domservb/notesdata
drwxr-xr-x   11 domservb users        4096 Sep 30 16:29 /domservb/notesdata
# chgrp notes /domservb/notesdata
# ls -ld /domservb/notesdata
drwxr-xr-x   11 domservb notes        4096 Sep 30 16:29 /domservb/notesdata
```

Create a profile with the following command:

```
echo “DOMINO_LINUX_SET_PARMS=1” >> ~domservb/.bash_profile
env | grep DOMINO
DOMINO_LINUX_SET_PARMS=1
```

Verify that you have enough disk space available to install the server. The Domino server we installed required 870 MB for the executables and 480 MB for the notesdata directory. We also needed 650 MB for the .tar file. The requirements of the release you install may differ slightly. Note that this is only for the minimum installation; for your environment you will need more, according to the size of your production data. Type the command **df -h** to display your available diskspace.

Ensure that you have connectivity from your workstation to the Linux server by issuing a ping command to the server IP name: For example:

```
ping domservb.itso.company.com
```

If this is not the first server installation, make sure that you have connectivity to the domain administration server that will supply the Domino directory.

You will need a Lotus Notes 6 administrator client with the remote server setup program in order to set up your Domino server.

## Change the default Linux kernel values

Some kernel files contain default kernel parameters that should be modified for the Domino environment. Note that these are system-wide values; there is no mechanism for changing them specifically for the Domino server application. These values are recommended based on experience in scaling Domino for Linux on zSeries with large numbers of mail users in a lab environment, finding bottlenecks, and tuning these kernel parameters to overcome them.

### File descriptors

By default, the Linux kernel limits the number of file descriptors that any one process can open; the default is 1024. File descriptors are used not only for open databases, but also for tcpip sockets that are opened when clients connect to the server. This limit affects Domino scalability, as it limits the number of users who can be connected to the Domino server. The combined total of open databases and client connections is limited to 1024. This default can be overridden by modifying the file /etc/security/limits.conf. Edit /etc/security/limits.conf as root and add or modify these lines:

```
*     soft   nofile    20000
*     hard   nofile    49152
```

where the “soft” value specifies the new default file descriptor limit (ulimit -n), and the “hard” value specifies the maximum file descriptor limit that may be set by the user.

**Tip:** If you change these limits by using an asterisk, as described above, the limits apply to *all* users, and if you are using SuSE SLES8 SP2 you will no longer be able to log in using ssh. (This problem is fixed in SP3.) You can instead add lines to this file for individual Linux users. For example, our Domino server Linux account ID is domservb and we changed the /etc/security/limits.conf as follows:

```
domservb soft nofile 20000
domservb hard nofile 49152
```

The users must login after /etc/security/limits.conf is saved in order for the change to take affect.

In addition to increasing the limit in /etc/security/limits.conf you will need to make another change, depending on how you start your Domino server.

1. Domino is always started manually, by logging in on the server id

   In order for the limits to be respected following logon, ensure the following line is in the appropriate **pam.d** login file:

```
session required/lib/security/pam_limits.so
```

For example, if **ssh** login is used, this line must appear in **/etc/pam.d/sshd**
If **telnet** is used, this line must appear in **/etc/pam.d/login**
If **rlogin** is used, this line must appear in **/etc/pam.d/rlogin**

The Domino server Linux account user must login after the above changes are made and saved.

2. Domino is automatically started through /etc/init.d

   Refer to the discussion in 9.5, “Automating Domino startup and shutdown” on page 239 for a sample script. It is worth noting, however, that /etc/security/limits.conf settings are not respected when Domino is automatically started by a /etc/init.d script. Therefore, a line such as the following must appear in the “start” section of such a script:

```
ulimit-n20000
```

No matter which mechanism is used, ensure the file descriptor limit has been successfully overridden for each of your Domino servers. One way to do that is by entering the following command on your Domino server console(s), and ensuring the result is “20000” (or higher):

load /bin/ksh -c “ulimit -n”

## /proc filesystem

The /proc filesystem contains several other parameters which should be increased for a production Domino server of any size. /proc is a virtual filesystem which is actually a window into kernel memory. Many of the files under /proc directories exist only to view and or set kernel parameters.

Table 7-3 lists the parameters to change, the recommended value, default value, and a description of the parameter. The recommended value can be set from a root user terminal session. Enter each setting as:

```
echo recommended-value > parameter-file-name
```

For example:

```
echo 131072 > /proc/sys/fs/file-max
```

These parameters can also be set with an environment variable as noted below.

*Table 7-3 Linux kernel settings*

| Parameter | Recommended value | Default value | Description |
|---|---|---|---|
| proc/sys/fs/file-max | 131072 | 196608 | Maximum number of file handles allowed for each process |
| proc/sys/net/ipv4/tcp_fin_timeout | 15 | 60 | Time to hold a socket in FIN-WAIT-2 state if it is closed by Domino |
| /proc/sys/net/ipv4/tcp_max_syn_backlog | 16384 | 1024 | Maximum number of connection requests that are remembered, but have not received acknowledgment from the connecting client |
| /proc/sys/net/ipv4/tcp_tw_reuse | 1 | 0 | Allow reuse of TIME-WAIT sockets |
| /proc/sys/net/ipv4/ip_local_port_range | 1024 65535 | 32768 61000 | Expand the range of port values |

Unfortunately, these changes are not permanent; the /proc filesystem is reloaded with default values during each boot. In order to avoid (re)setting /proc parameters after each boot,

Domino will set them for you at Domino startup, but only if the environment variable DOMINO_LINUX_SET_PARMS is set to “1”, as in:

```
export DOMINO_LINUX_SET_PARMS=1
```

Set this variable in the .profile of the Domino server Linux account.

**Note:** If you plan to start your Domino server at Linux boot time or automatically using a script, make sure the values above are set properly.

## 7.4 Installing the executable server code

This section covers the following:

- Placing the tar file on your Linux filesystem
- Extracting the files
- Running the installation script in interactive mode

**Important:** All steps described in this section must be executed from root (UID = 0).

### 7.4.1 Placing the tar file on your Linux filesystem

Transfer the Domino Server tar file from the CD-ROM on your workstation using ftp to the Linux filesystem so it can be decompressed. The tar file must be transferred in binary mode. The target directory for the ftp should be /opt/lotus.

### 7.4.2 Extracting the files

To extract the tar file, use the tar command:

```
tar -xvzf <filename>
```

For example:

```
tar -xvf Zlinux650000prod.TAR
zlinux/
zlinux/sets/
zlinux/sets/fgroups/
zlinux/sets/fgroups/fg0.taz
zlinux/sets/fgroups/fg1.taz
zlinux/sets/fgroups/fg2.taz
.....
```

This will unpack the file and place the packed files in the correct paths below the /opt/lotus path. Depending on your machine and workload, this can take a couple of minutes.

Enter the command **`ls -al`** to validate that the subdirectory zlinux exists. You can also check the return code from the tar command by typing **`echo $?`**. If the return code equals zero, you should be fine.

After you finished the installation and setup of your Domino server and you are satisfied that you will not need the tar file, you might want to delete it. Change to /opt/lotus and enter:

```
rm server_tar_file
```

## 7.4.3 Running the installation script

To install the Domino program files on your system, you can use either interactive mode or script mode.

Using the interactive mode, the Install Program will prompt you for all settings necessary to install the Domino Server. If this is your first installation of Domino for Linux on zSeries, we recommend that you use the interactive mode.

With the scripted mode, you enter the values in a script file. The installation process will get the values from the file and run the installation. For instructions on how to install with a script, refer to 7.7, “Installing the server executables using Script mode” on page 166. If you will be installing the Domino server more than once (due to errors, testing, or multiple servers), the script mode is very useful.

### Interactive mode

Here we discuss using interactive mode to install the Domino program and data files on your Linux system. During the installation, you can use these keys at the command prompt:

- Type: `h` for help.
- Type: `e` to exit the Install program.
- Press Esc to return to the previous screen.
- Press the spacebar to change the setting until you get the one you want.
- Press Tab to accept a setting and continue to the next screen.

To run the interactive install, make sure you are in the /opt/lotus/zlinux directory and enter: `./install` as shown in Figure 7-1.

```
root@linuxa:/opt/lotus/zlinux > ls -al
total 64
drwxrwx---    4 root     notes         184 Aug  5 18:52 .
drwxr-xr-x    5 root     root          264 Aug  6 14:30 ..
-r-xr-xr-x    1 95501    notes        4921 Apr 11 12:36 install
-r--r--r--    1 95501    notes       39051 Mar 20 20:06 license.txt
-r--r--r--    1 95501    notes       12995 Apr 27 20:06 script.dat
dr-xr-xr-x    4 root     notes          96 May 21 20:10 sets
dr-xr-xr-x    4 root     notes          96 May 21 20:11 tools
root@linuxa:/opt/lotus/zlinux >./install
```

*Figure 7-1 Running the install script*

1. The ./install tells the operating system to look in the current directory for the executable named install. For security reasons, ./ is not added to the root PATH environment variable, since you could be tricked into launching a malicious program from a current directory, such as the /tmp folder.

The ./install command will bring up the welcome page, as shown in Figure 7-2.

```
==========================================================================================
                                Domino Server Installation
==========================================================================================

    Welcome to the Domino Server Install Program.


    Type h for help on how to use this program.
    Press TAB to begin the installation.


------------------------------------------------------------------------------------------
Type h for help
    Type e to exit installation
    Press TAB to continue to the next screen.
------------------------------------------------------------------------------------------
```

*Figure 7-2 Welcome page*

Press Tab to continue. This will take you to Figure 7-3.

```
==============================================================================
                         Domino Server Installation
==============================================================================

    A lot of new features have been added to the Domino  Server.
    In order to install your server correctly, please read the Domino
    Server release notes first, then run your installation. Otherwise,
    you may experience problems when using the new features.



------------------------------------------------------------------------------
    Type e to exit the Install program.
    Press ESC to return to the previous screen
    Press TAB to continue to the next screen.
------------------------------------------------------------------------------
```

*Figure 7-3 First installation screen*

2. Keep pressing Tab till you get through the many license agreement screens to the screen shown at Figure 7-4 on page 141.

```
================================================================================
                          Domino Server Installation
================================================================================


    You may proceed with the installation only if you agree to the
    terms and conditions of the Lotus Domino Software Agreement.


--------------------------------------------------------------------------------
    Type e to exit the Install program.
    Press ESC to return to the previous screen.
    Press the Spacebar to change the setting until you get the one you want.
    Press TAB to accept a setting and continue to the next screen.
--------------------------------------------------------------------------------

>>> Do you agree to the terms of the license agreement ?  [Yes]
```

*Figure 7-4 License agreement screen*

3. Select **Yes** and press Tab to continue. Typing: `No` takes you back to the beginning of the license agreement. There you can choose to end the installation or the re-read the agreement. Typing: `Yes` will bring up the screen shown in Figure 7-5.

```
================================================================================
                          Domino Server Installation
================================================================================

    Here you are given the option to install additional or upgrade
    existing Server Partitions. Selecting "Yes" below will allow
    you to specify which Data directories to install. Only those
    Data directories specified will be upgraded or added.

    The existing Program directory must be specified  in order for
    new Server Partitions to be created. However,  existing Data
    directories do not need to be listed. Any existing Data
    directories that are listed will be installed to, and old templates
    in those Partitions will be overwritten.

    If you wish to add more than  one Partition  to  your existing
    Domino server, select "Yes" when asked if you want to run
    multiple server partitions on this system.  Otherwise you will
    only be able to upgrade or install one Data directory.

    Warning:
      If you do not have an existing Domino Server on your system,
    please select "No" for the option to add data directories only.

--------------------------------------------------------------------------------
    Type e to exit the Install program.
    Press ESC to return to the previous screen.
    Press the Spacebar to change the setting until you get the one you want.
    Press TAB to accept a setting and continue to the next screen.
--------------------------------------------------------------------------------

>>>Do you want to install data directories only? [No ]
```

*Figure 7-5 Install data directory only screen*

4. At this point, you make a basic decision on what will be installed. You may install the Domino server executable code and Domino server (as we did), upgrade the Domino server executable code, or simply add a new partitioned server to an existing Domino installation. Adding a new partitioned server creates an additional notesdata directory and populates it with Domino templates and databases, but does not change the Domino server code. Choose one:

   **Yes** To add notesdata directories to existing Domino servers. Use this option if you have already installed Domino and want to install additional Domino partitions. For more information about setting up more then one Domino server, refer to 7.9, “Installing multiple Domino versions” on page 168

   **No** Installs or upgrades the Domino server code.

   Since we want to install Domino server code, we accepted the default of No.

   Press Tab with the **[No]** option selected. Proceed to the select server type screen shown in Figure 7-6.

```
================================================================================
                         Domino Server Installation
================================================================================


    Select the type of installation you want.


--------------------------------------------------------------------------------
    Type h for help.
    Type e to exit the Install program.
    Press ESC to return to the previous screen.
    Press the Spacebar to change the setting until you get the one you want.
    Press TAB to accept a setting and continue to the next screen.
--------------------------------------------------------------------------------

>>> Select Setup type : [Domino Enterprise Server]
```

*Figure 7-6 Choose server type*

5. Select the type of server you want to install.

   **Domino Utility Server** - This installs a Domino server that provides application services only, with support for Domino clusters. The Domino Utility Server is a new installation type for Lotus Domino 6 that removes client access license requirements. Note that it does *not* include support for messaging services; refer to the full licensing text for details.

   **Domino Messaging Server** - This installs a Domino server that provides messaging services. Note that it does *not* include support for application services or Domino clusters.

   **Domino Enterprise Server** -This installs a Domino server that provides both messaging and application services, with support for Domino clusters.

   **Note:** All three types of installations support Domino partitioned servers. Only the Domino Enterprise Server supports a service provider (ASP) environment.

   For our installation, we chose the Domino Enterprise Server. Press Tab to select [Domino Enterprise Server] and go on with Figure 7-7 on page 143.

```
================================================================================
                          Domino Server Installation
================================================================================


     The optional installation feature for template files is designed for
     users who are installing over a previous version of the Domino Server
     and wish to keep all previous template files. If this is not an
     installation over an existing Domino Server, all template files must
     be installed.

     Warning:
            To ensure proper operation of your Domino Server, we highly
     recommend installing all template files.  Only select [No] if you are
     an advanced user and you know that this server already has the latest
     template files.  The Domino Server will not run properly without the
     latest templates.

--------------------------------------------------------------------------------
    Type e to exit the Install program.
    Press ESC to return to the previous screen.
    Press the Spacebar to change the setting until you get the one you want.
    Press TAB to accept a setting and continue to the next screen.
--------------------------------------------------------------------------------

>>>Do you want to install all template files for this Domino Server? [Yes]
```

*Figure 7-7 Template installation*

6. With Domino 6, you have a new option: to install a subset of the templates provided with the server. If you already are running Domino and have customized the templates and do not want to overwrite your templates, then choose No.

   In our case, however, this was a new installation so we installed all the templates. In general, you will need all templates in order to take advantage of new features and fixes. Choose one:

   **Yes** To install new templates

   **No** To retain templates from a previous release

   Press Tab to install all the templates and to continue with the screen about ASP functionality in Figure 7-8 on page 144.

```
================================================================================
                          Domino Server Installation
================================================================================


    The option to setup an ASP server refers to the configuration of an
    Application Service Provider server. This type of server can only be
    configured after an Enterprise Server installation.
    Selecting "Yes" below will cause the Domino Setup program to configure
    the server appropriately for ASP functionality. This will add security
    features not present in a normal configuration, so do not select "Yes"
    unless an ASP configuration is specifically required for this server.

    The default value is "No", which is recommended for performing server
    upgrades and/or non-ASP installations.

--------------------------------------------------------------------------------
    Type e to exit the Install program.
    Press ESC to return to the previous screen.
    Press the Spacebar to change the setting until you get the one you want.
    Press TAB to accept a setting and continue to the next screen.
--------------------------------------------------------------------------------

>>>Do you want to configure this server with ASP functionality? [No ]
```

*Figure 7-8 ASP functionality*

7. Here you have the option to set up an Application Service Provider (ASP) Domino server. Many of the ASP functions are also available in a standard Domino configuration. The ASP enhancements better facilitate an ASP environment.

   Press Tab and select **[No]**, because an Enterprise Server has not yet been installed.

8. During the next steps (through step 15 on page 148), you define where the Domino program code (executable files) and the notesdata directory will be placed. Use your worksheet from 7.2, “Checklist” on page 134 as a guide.

   For our installation, we have chosen a filesystem as illustrated in Figure 4-6 on page 73. The executables were installed in /opt/lotus (as shown in Figure 7-9 on page 145), and the notesdata directory under /domservb/notesdata.

```
================================================================================
                          Domino Server Installation
================================================================================


    The program directory is the path where the Install program
    installs the Domino program files.  The Install program
    automatically adds "lotus" to the path.


--------------------------------------------------------------------------------
    Type h for help.
    Type e to exit the Install program.
    Press ESC to return to the previous screen.
    Press ENTER to edit a setting.
    Press TAB to accept a setting and continue to the next screen.
--------------------------------------------------------------------------------

Current program directory setting : /opt/lotus
```

*Figure 7-9 Program directory*

9. Specify the directory into which Domino program files will be stored. We chose to install the server in the default program directory /opt/lotus.

   Press Tab to continue on to Figure 7-10.

```
================================================================================
                          Domino Server Installation
================================================================================


    You will now be prompted for information on how to install one or
    more Domino Data Directories.

    Please note that the UNIX user and group names asked for will own
    all of the data directories specified.

    The system will own the program files.


--------------------------------------------------------------------------------
    Type e to exit the Install program.
    Press ESC to return to the previous screen
    Press TAB to continue to the next screen.
--------------------------------------------------------------------------------
```

*Figure 7-10 Information screen for installing the Domino data directory*

10. Figure 7-10 outlines the basic file ownership concept of Domino running on Linux. The user and group you specify will own the data and will be used to launch the server. You need to associate the Linux user IDs you defined in 7.3, “Pre-installation steps” on page 135 to the Domino server you are installing.

    Press Tab to continue to Figure 7-11 on page 146.

11. In Figure 7-11 on page 146 you are asked if you want to set up partitioned servers. These are servers that use the same Domino executable code but have separate data directories. If you plan to install multiple Domino servers, see 7.8, “Adding a partitioned

Domino server (DPAR)" on page 167 and 7.9, "Installing multiple Domino versions" on page 168.

Press Tab to select **[No]** and continue.

.

```
==============================================================================
                        Domino Server Installation
==============================================================================


    You can run more than one Domino Server on a single computer
    at a time based on this installation.  This feature is called
    Domino Partitioned Servers, and requires separate Data Directories
    for each Domino Server to be run.


------------------------------------------------------------------------------
    Type h for help.
    Type e to exit the Install program.
    Press ESC to return to the previous screen.
    Press the Spacebar to change the setting until you get the one you want.
    Press TAB to accept a setting and continue to the next screen.
------------------------------------------------------------------------------

>>>Do you want to run more than one server based on this installation?  [No ]
```

*Figure 7-11 Domino partitioned server*

## One Domino partition installation

Figure 7-12 asks for the location of the Domino data directory. If you want to change the default path for notesdata directory, press Enter to edit the setting. In our installation, we chose to install it in /domservb/notesdata.

Press Tab to continue with Figure 7-13 on page 147.

```
==================================================================================
                            Domino Server Installation
==================================================================================


    The data directory is the path where the Install program
    installs the Domino data files.


----------------------------------------------------------------------------------
    Type h for help.
    Type e to exit the Install program.
    Press ESC to return to the previous screen.
    Press ENTER to edit a setting.
    Press TAB to accept a setting and continue to the next screen.
----------------------------------------------------------------------------------

Current data directory setting : /local/notesdata
```

*Figure 7-12 Choosing notesdata location*

```
==========================================================================================
                              Domino Server Installation
==========================================================================================


    Please enter the Domino UNIX user name.  This UNIX user will own the
    Domino data files, and be used to run the Domino Server.

    NOTE for the upgrade installer:
    The Domino UNIX user name/account name you specify here must be the same
    as the owner of the existing installed data files for proper operation
    of Domino.


------------------------------------------------------------------------------------------
    Type h for help.
    Type e to exit the Install program.
    Press ESC to return to the previous screen.
    Press ENTER to edit a setting.
    Press TAB to accept a setting and continue to the next screen.
------------------------------------------------------------------------------------------

Current UNIX user setting : notes
```

*Figure 7-13 Choosing a Linux user ID for your Domino server*

12. This is the Domino user name on Linux. If you have multiple Domino server partitions, you will have multiple Domino server Linux accounts and each one will be used to start a Domino server. This user account also owns the data directory. For our environment, we used domservb.
13. In Figure 7-14 on page 148, you have to decide what Linux group the Domino server belong to. Proceed by pressing Tab.

.

```
==========================================================================================
                              Domino Server Installation
==========================================================================================

    Please enter the Domino UNIX group.  This UNIX group will own the
    Domino data files.  The Domino UNIX user must be a member
    of this group.

    NOTE for the upgrade installer:
    Domino UNIX group/account group you specify here must be the same as
    the owner of the existing installed data files for proper operation
    of Domino.

------------------------------------------------------------------------------------------
    Type h for help.
    Type e to exit the Install program.
    Press ESC to return to the previous screen.
    Press ENTER to edit a setting.
    Press TAB to accept a setting and continue to the next screen.
------------------------------------------------------------------------------------------

Current UNIX group setting: notes
```

*Figure 7-14 Choosing a Linux group for your Domino server user ID*

14. As discussed in 7.3, “Pre-installation steps” on page 135, we recommend you create a group to which all your Domino Linux user accounts belong. In our environment, we kept the default name *notes*.

15. Select server setup method. Figure 7-15 on page 149 asks you how the server setup program is invoked.

    **Remote Server Setup** This option starts the server in listen mode so that you can connect to it with the Domino server setup program. We recommend using this method, since it starts the server startup with -listen. It makes it easier to start the setup.

    **Manual Server Setup** Use this option if you want to do the rest of the installation at a later time. You will have to start the server manually by logging in as the Domino server user (domservb), then go to your notesdata directory and start the server manually with:

    ```
    /opt/lotus/bin/server -listen
    ```

    **Local Server Setup** This option is not available for Domino on zSeries. There is no local terminal session available for the Java-based interactive dialog.

```
==========================================================================================
                              Domino Server Installation
==========================================================================================

    This optional feature is for single local Domino Server installations only.
    For new server installations, server setup will be launched automatically
    after the installation is complete.  For upgrade installations, the server
    will be restarted automatically after the installation is complete.

    The default value is "Manual Server Setup" which does not launch server setup
    or restart the server after installation. The installation will  complete and
    you will then have to launch server setup or restart the server manually.

    Select "Local Server Setup" to launch server setup after a new server
    installation or to restart the server after a server upgrade.

    Select "Remote Server Setup" to launch server setup in listen mode for new
    server installations. You will then be able to connect to the server with the
    Remote Server Setup tool.


------------------------------------------------------------------------------------------
    Type e to exit the Install program.
    Press ESC to return to the previous screen.
    Press the Spacebar to change the setting until you get the one you want.
    Press TAB to accept a setting and continue to the next screen.
------------------------------------------------------------------------------------------

>>>Select server setup method: [Remote Server Setup]
```

*Figure 7-15 Select server setup method*

Press the spacebar until [Remote Server Setup] appears, then press Tab to go to Figure 7-16.

```
==============================================================================
                        Domino Server Installation
==============================================================================


    Your configuration of the Install program is complete.

    By continuing, the Install program will first allow
    you to review your configuration settings before
    beginning the installation.


------------------------------------------------------------------------------
    Type e to exit the Install program.
    Press ESC to return to the previous screen
    Press TAB to continue to the next screen.
------------------------------------------------------------------------------
```

*Figure 7-16 First part on installation complete*

16. You have now finished responding to the prompts of the interactive install dialog. If you had used the script mode, you would have placed these responses into the script.

Press Tab to continue and to review your input, as shown in Figure 7-17.

```
================================================================================
                          Domino Server Installation
================================================================================
Installation settings:

    Installation type    : Domino Enterprise Server

  Install template files :  Yes
  Server Setup Method    :  Remote Server Setup
  Configure to ASP Server:  No

    Program directory    : /opt/lotus
    Data directory       : /domservb/notesdata
      UNIX user          :  domservb
      UNIX group         :  notes


Press the Escape key to re-configure the settings
  or
Press the Tab key to perform the installation...
```

*Figure 7-17 Verify your settings*

17. Review this panel carefully to see if everything is as you planned it. If you find anything to change, press Esc to reconfigure the settings.

    If everything looks alright, press Tab to begin the installation. Figure 7-18 on page 151 is the last installation screen on Linux. Everything from there on can be configured using the remote server setup and the Domino administrator client on your workstation.

```
Validating...


For the latest patch DB please go to http://www.lotus.com/ldd/checkos


This will check the Operating System level and tell you what is missing. Note, no patch
list if all patches are present.




The OS appears to have the correct patches .
Installing Domino Server kits ...
The Domino Server installed successfully.

Configuring Domino Server from
Unix user name:  domservb
Unix group name: notes
Domino program directory: /opt/lotus
Domino data directory: /domservb/notesdata
/proc/2841/mapped_base has been set to the recommended value of 134217728
The value in /proc/sys/fs/file-max is 26214, which does not match the recommended value
of 131072
The value in /proc/sys/net/ipv4/tcp_fin_timeout is 60, which does not match the
recommended value of 15
The value in /proc/sys/net/ipv4/tcp_max_syn_backlog is 1024, which does not match the
recommended value of 16384
The value in /proc/sys/net/ipv4/tcp_tw_reuse is 0, which does not match the recommended
value of 1
The value in /proc/sys/net/ipv4/ip_local_port_range is 32768    61000, which does not
match the recommended value of 1024        65535
./java -ss512k -Xoss5M -cp jhall.jar:cfgdomserver.jar:Notes.jar
lotus.domino.setup.WizardManagerDomino -data /domservb/notesdata -listen
Remote server setup enabled on port 8585.

The Domino setup server is now in listening mode.
A remote client can now connect to this server and configure Domino.

To connect to this server, launch the Remote Domino Setup program from a command-prompt
as follows:
From a Domino administrator client: serversetup -remote
From a Domino server: server -remote

To end this server, launch the Remote Domino Setup program from a command-prompt as
follows:
From a Domino administrator client: serversetup -q linuxb
From a Domino server: server -q linuxb

For more information, see the printed guide Setting Up Domino Networks and Servers.
```

*Figure 7-18 Starting the server in listen mode*

18. The installation program then validates your entries and does a check of the operating system to see if your Linux is at the correct level.

**Important:** Check the readme file to see how CheckOS will function on your version of Domino. Up through 6.5.3 Domino for Linux on zSeries did not use the CheckOS feature, so the message that the OS appears to have the correct patches is not verified.

19. Look for the message: `The Domino Server installed successfully.` on your screen, somewhere in the midst of the other messages. This confirms that you have finished installing the server executable files onto your system. The next step is to set up your Domino server.

Notice the messages in Figure 7-18 on page 151 about not having the correct kernel values. Refer to 7.3, “Pre-installation steps” on page 135 for instructions on how to set the correct values

If you receive an error message, you will need to fix it, then re-run the installation from the start. A typical error message involves either incorrectly specifying the user or group, or else failing to create the user or group before beginning the installation.

Another common error message concerns lack of disk space. Refer to 7.10, “Avoiding installation problems” on page 170 for more information about what to do if you have a problem.

If you chose Remote Server Setup (see Figure 7-15 on page 149), the Domino server will start automatically in listen mode. It is ready for you to connect to it with the Domino server setup program and set up your server.

## 7.5 Setting up the Domino server

Before you start the Domino server setup, you should complete your Domino infrastructure planning. If this is your first Domino server, you will be prompted during server setup for the Domino server name, your Organization name, the name of your Domino administrator, and your Domino domain name. Do not proceed with server setup until you have created or collected this information. If you are planning a large Domino implementation, you should dedicate time for planning your Domino infrastructure, including Organization, Organizational Unit, and Domains.

The new Java setup allows for remote configuration. To run the remote setup, you must have installed the Domino Administrator client with the remote server setup option on your workstation. Figure 7-19 on page 153 shows where to install the remote server setup.

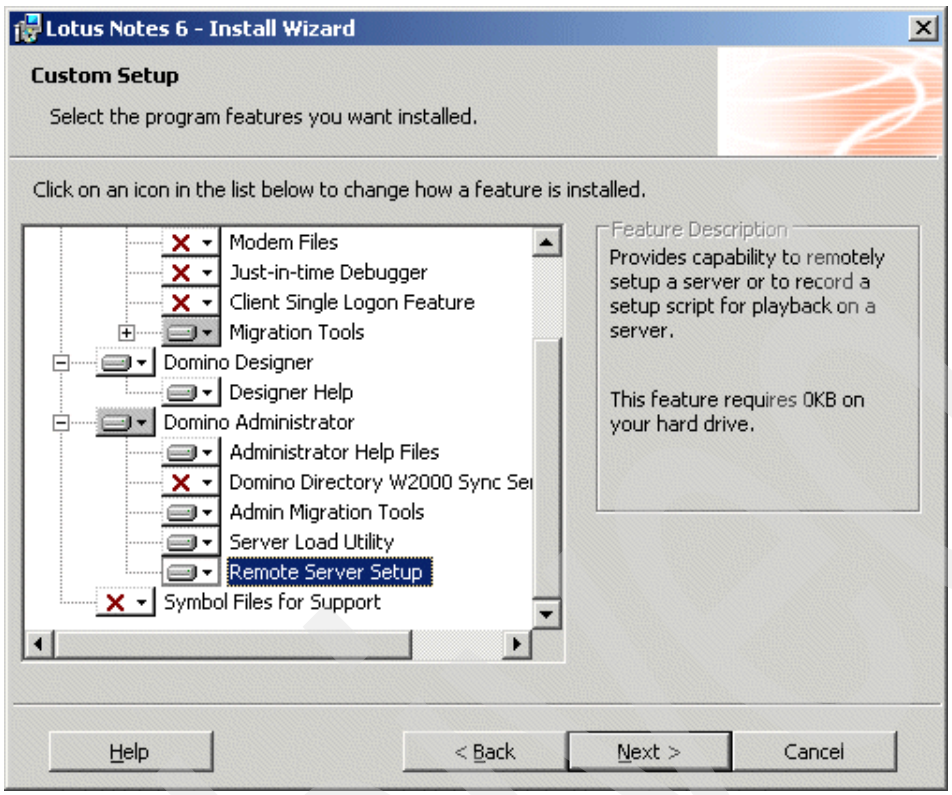



*Figure 7-19 Lotus Administrator remote server setup option*

20. If you chose manual server setup at step 15 on page 148, log on to the server with the Domino user account (domservb), change to the Domino data directory (/domservb/notesdata), and start the Domino server with the listen option as described. If you chose remote server setup, the server is already started in listen mode.

    If you are setting up multiple Domino partitions, you have to invoke the server setup for each partition you try to install separately.

21. Go to **Start Programs** → **Lotus Applications** → **Remote Server Setup** to start the Java based-server setup program, as shown in Figure 7-20.

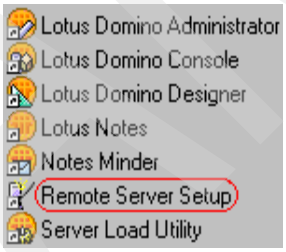



*Figure 7-20 Remote server setup*

**Tip:** The installation guide has a section on the Domino server setup containing information about how to use the program to automate your installation.

22. You will see a pop-up window, as shown in Figure 7-21 on page 154.


*Figure 7-21 Connect to remote server*

Enter either the server's name or IP address in the Remote Host Address field, then click **Ping** (see Figure 7-21) to ensure that you have connectivity.

Click **OK** to see the welcome screen displayed in Figure 7-22.

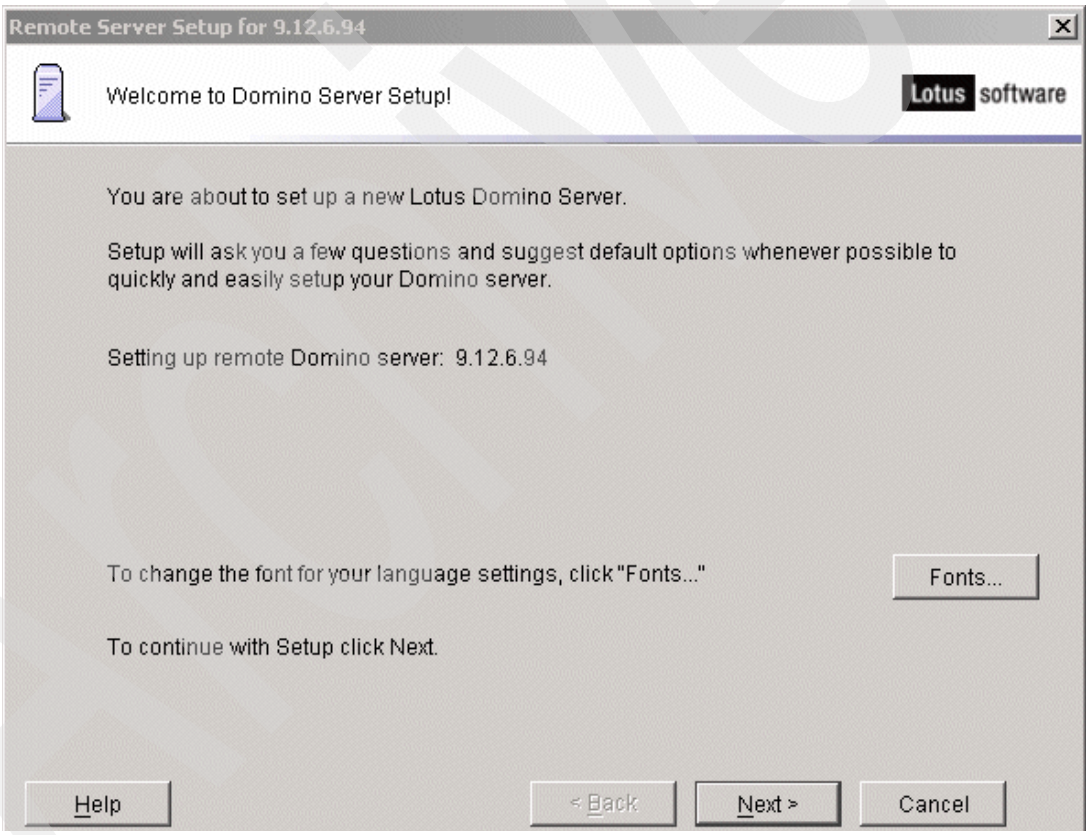



*Figure 7-22 Welcome screen of Domino server setup*

23. Click **Next** to continue. If the fonts on your screen do not look right, press **Fonts** to change the font. Then continue with Figure 7-23 on page 155.

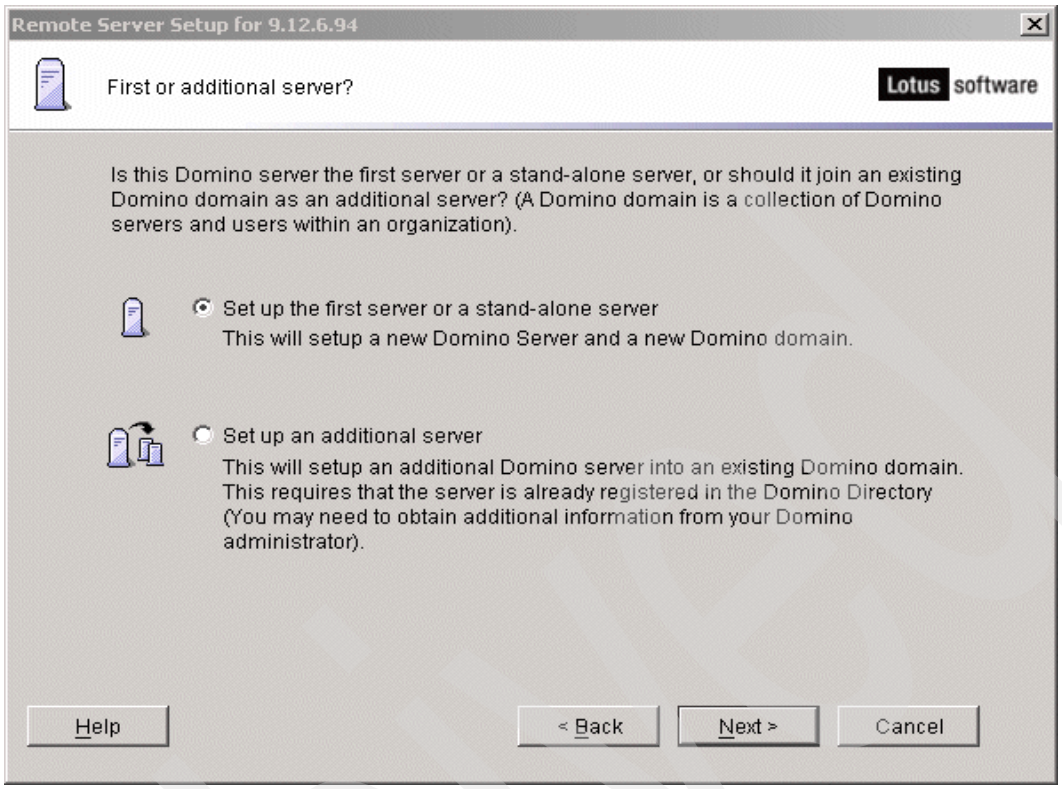



*Figure 7-23 First or additional Domino Server?*

We have set up the first server in our new ITSO domain. If you are setting up an additional server, you will be prompted to specify the location of your server ID and the hierarchical name of the additional server. Once you have done so, you can skip to Step 28 on page 160.

Click **Next** to continue with Figure 7-24 on page 156.

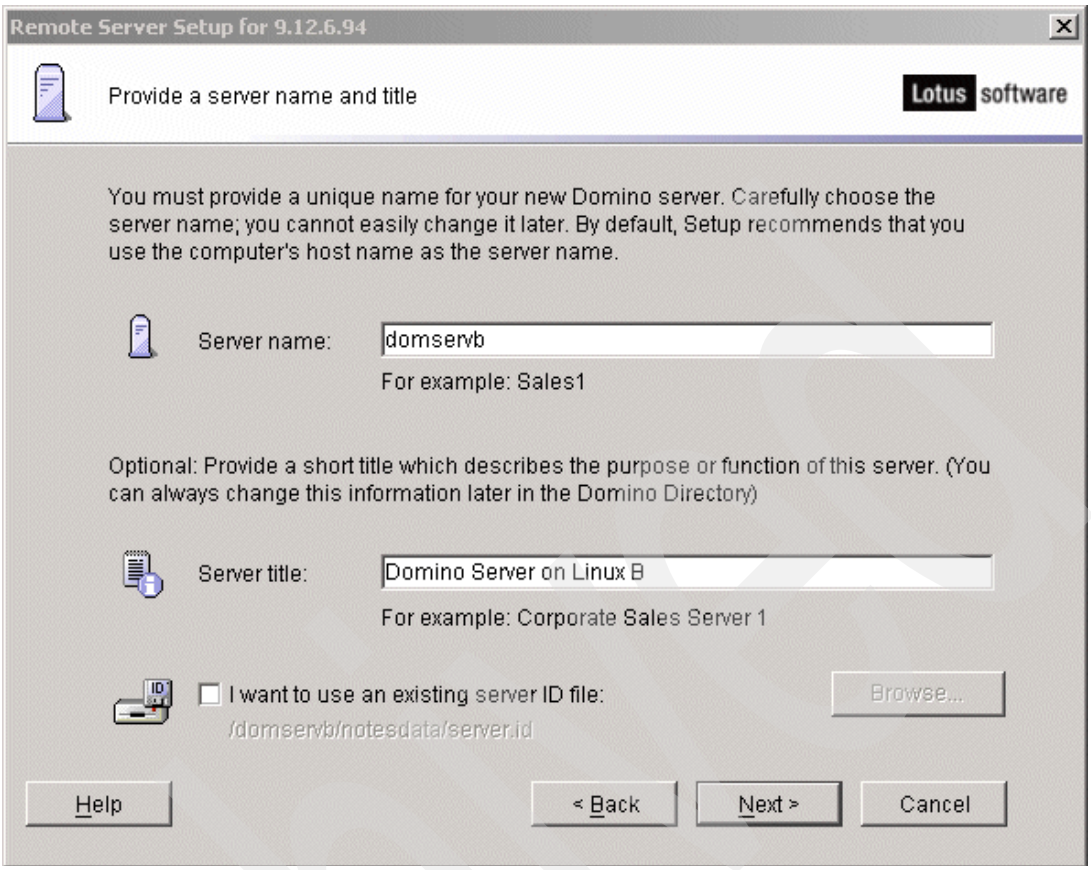



*Figure 7-24 Domino server name and title*

24. Provide a self-describing name for your Domino server. You might want to add this server name to your DNS. If you enter it as an alias for the host of your Domino server, any client will be able to resolve the IP address and therefore connect to the Domino server.

    The Server title gives you an opportunity to provide a short description of the server's main function or the organization to which it belongs.

    Click **Next** to continue with the naming of your Domino organization as shown in Figure 7-25 on page 157.

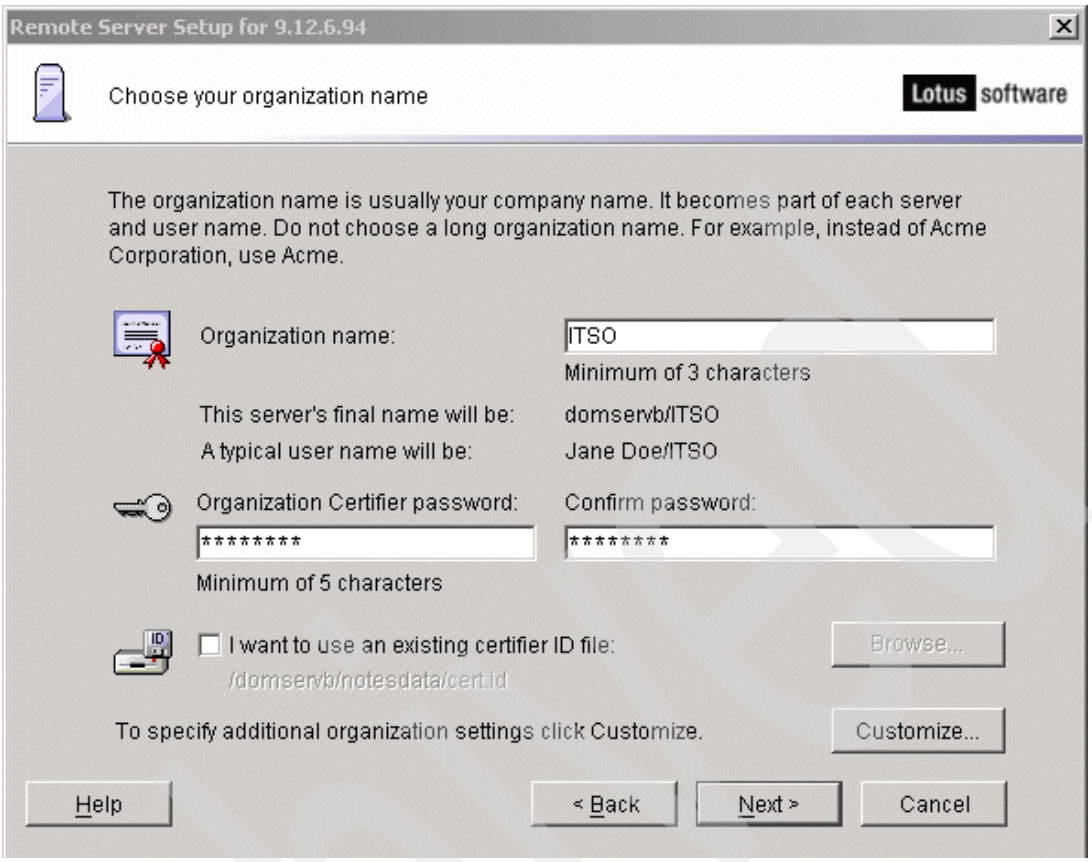



*Figure 7-25 Domino organization name*

25. Set a meaningful Organization name, and make certain to enter a secure password for your Certifier ID, then click **Next** to proceed with naming your Domino domain in Figure 7-26 on page 158.

    If you want to use an existing certifier, check the box: `I want to use an existing certifier ID file:`.

**Tip:** The Certifier ID is the key to all user and server authentication. It should be removed from the server's filesystem immediately after you have finished the setup and stored in a secure location. You should also rename the file (it will be named cert.id by default) to include the organization name.

Keep in mind, though, however, that you will need the Certifier ID in order to create subsequent Organizational Units (OUs). Additional OUs are useful for distinguishing people from servers, as well as distinguishing departments or regions. You should settle on a scheme that minimizes the number of OUs but provides sufficient detail. See *Domino 6 Administration Help* database for further details.

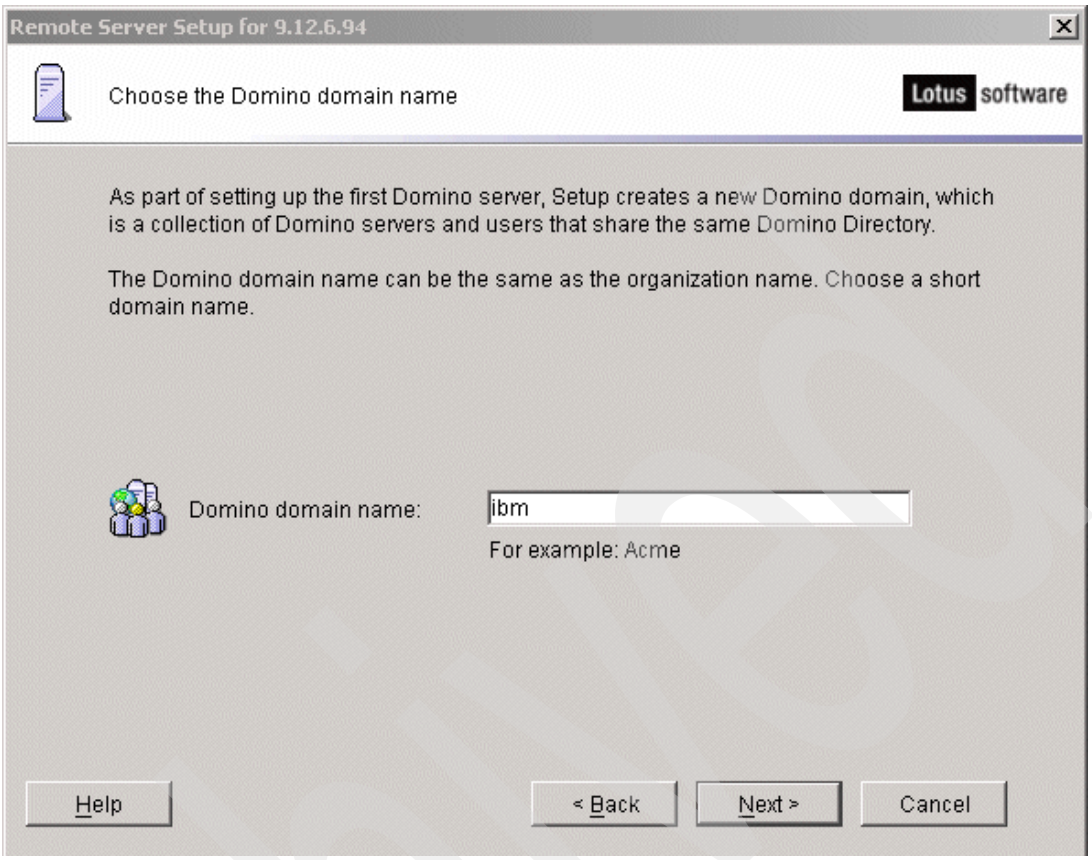



*Figure 7-26 Domino domain name*

26. If you plan to have multiple domains, use your naming scheme and make certain the first domain conforms to the scheme you will use for all subsequent domains.

    Type the name you want to use and click **Next**. In Figure 7-27 on page 159, you have to specify your Domino administrator for your new server.

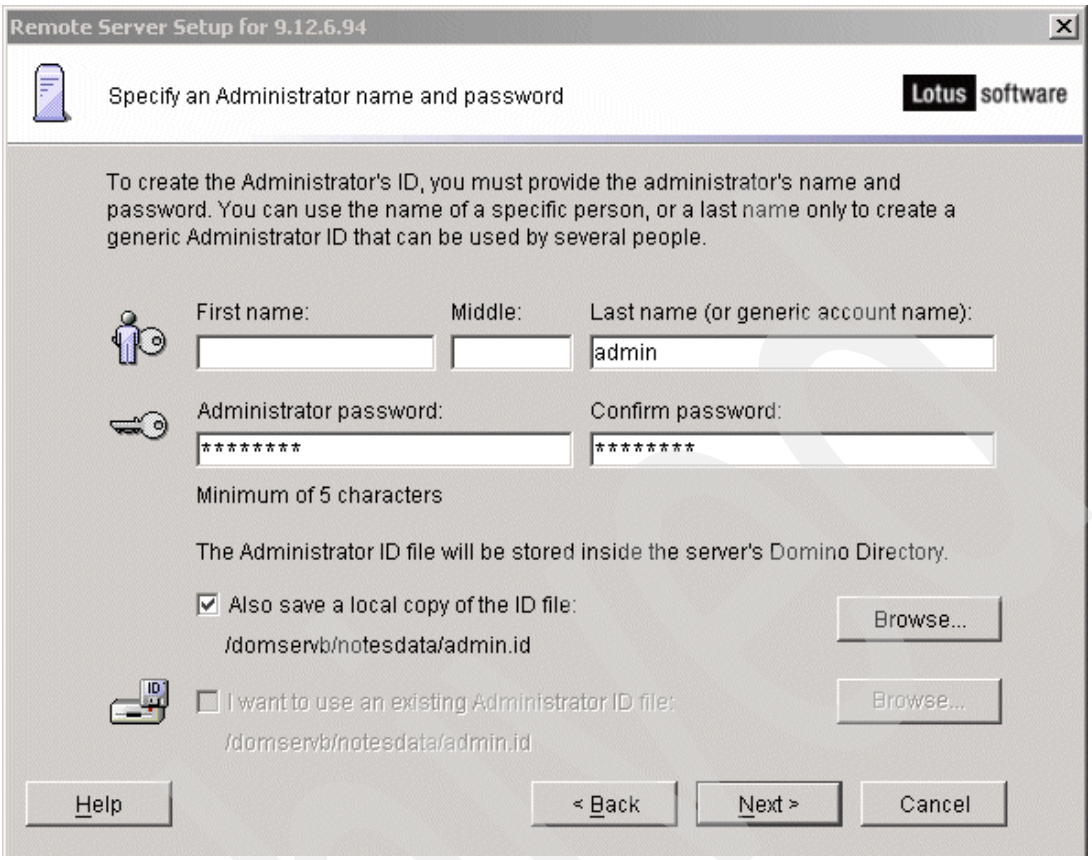



*Figure 7-27 Domino Administrator name and password*

27. Enter an administrator name and password, then click **Next** to move to Figure 7-28 on page 160.

    If you intend to use the ID locally, check the option: `Also save a local copy of the ID file:` so that you will have easy access to the ID. Since the Administrator ID has full access to the Domino Directory, it is recommended that you remove the ID from the Person document after downloading it.

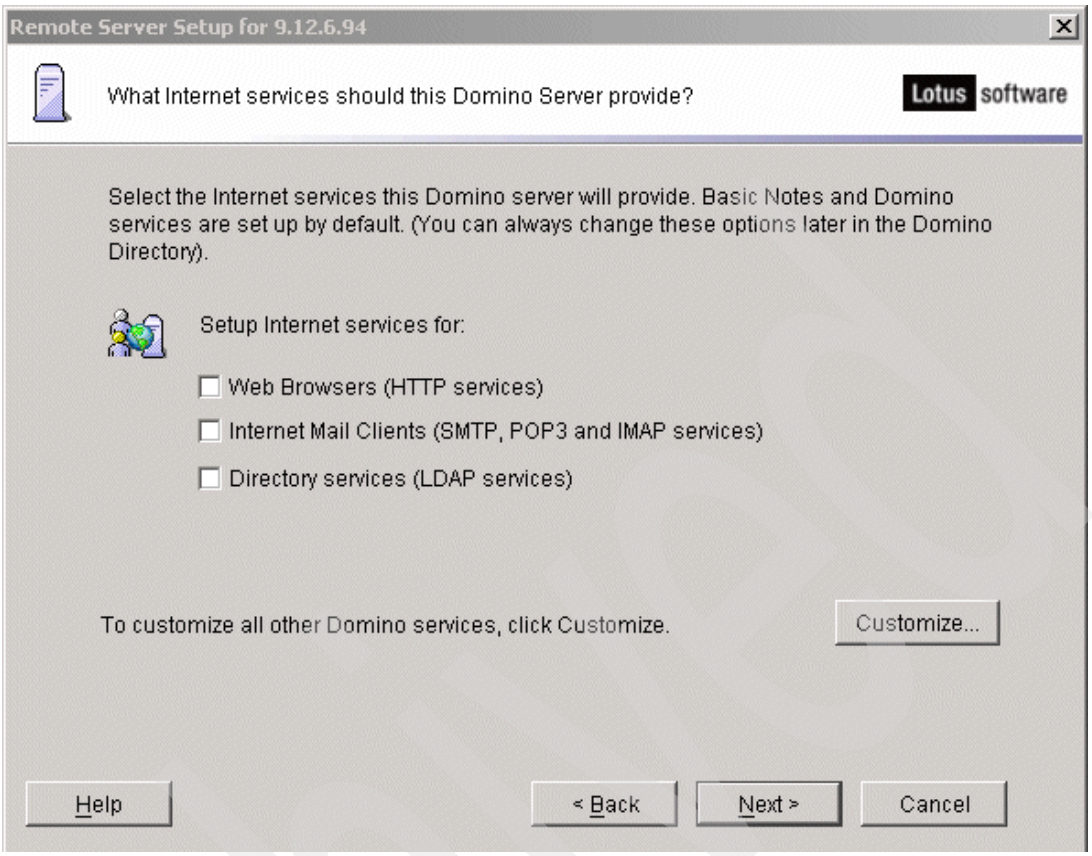



*Figure 7-28 Chose services*

28. If you do not need LDAP services, deselect the LDAP check box; you can always start LDAP and other services from the Domino console.

    Proceed with the network definitions in Figure 7-29 on page 161.

**Tip:** You can always add a service later by modifying the ServerTasks= line of the notes.ini or issuing a **`set config servertasks=`** command from the Domino console. With the **`set config`** command, you need to enter every service you want to have running, not just the ones to add. You can see the existing services by entering the command **`show config servertasks`**.

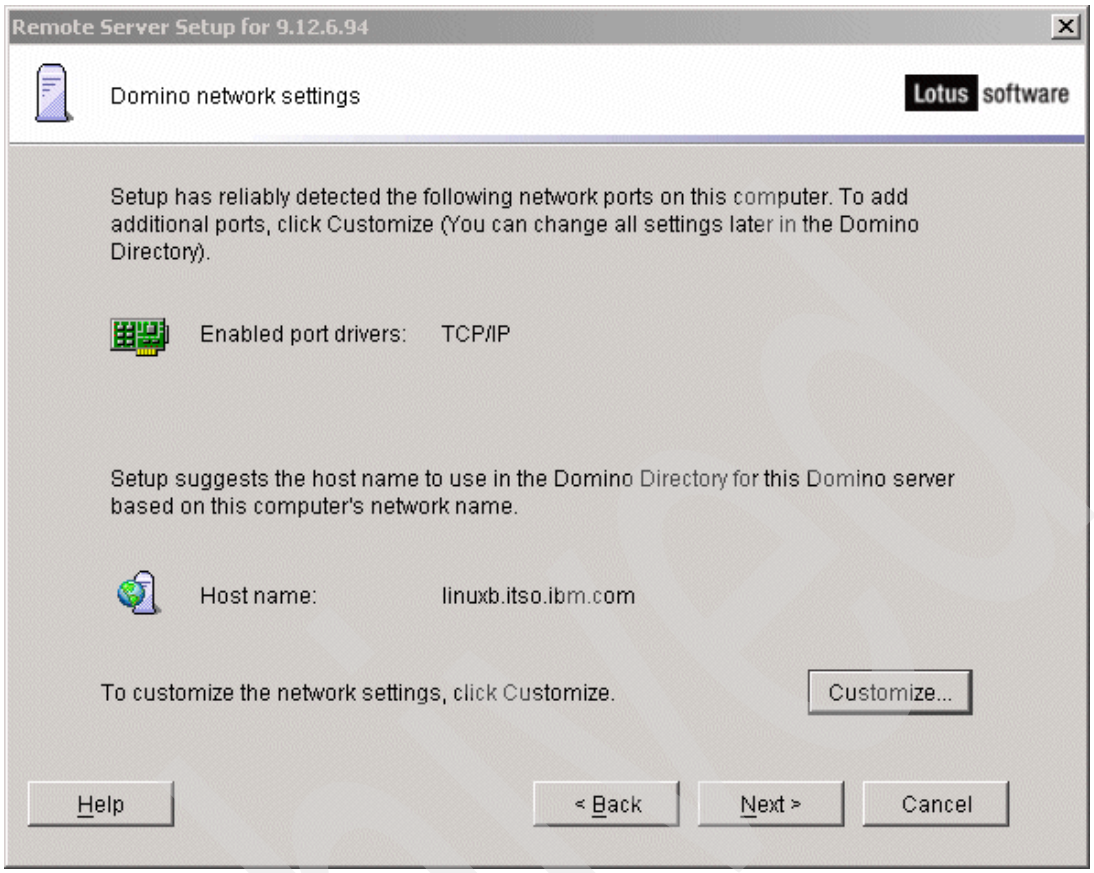



*Figure 7-29 Domino network settings*

29. The auto-detect should determine your network port and host name, as shown in Figure 7-29. If it did not, press Customize to enter the correct settings.

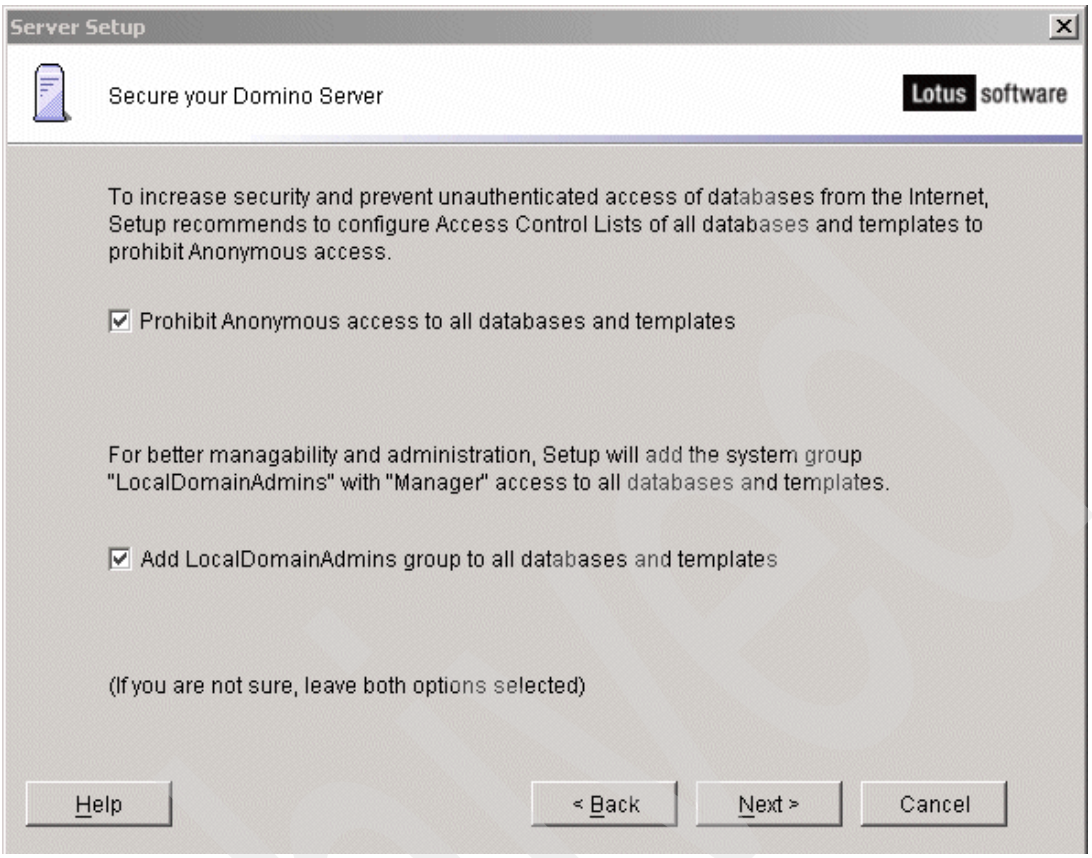



*Figure 7-30 Secure your Domino server*

30. To increase security, ensure that the two check boxes are marked (as you see in Figure 7-30) and click **Next** to proceed with the setup; see Figure 7-31 on page 163.

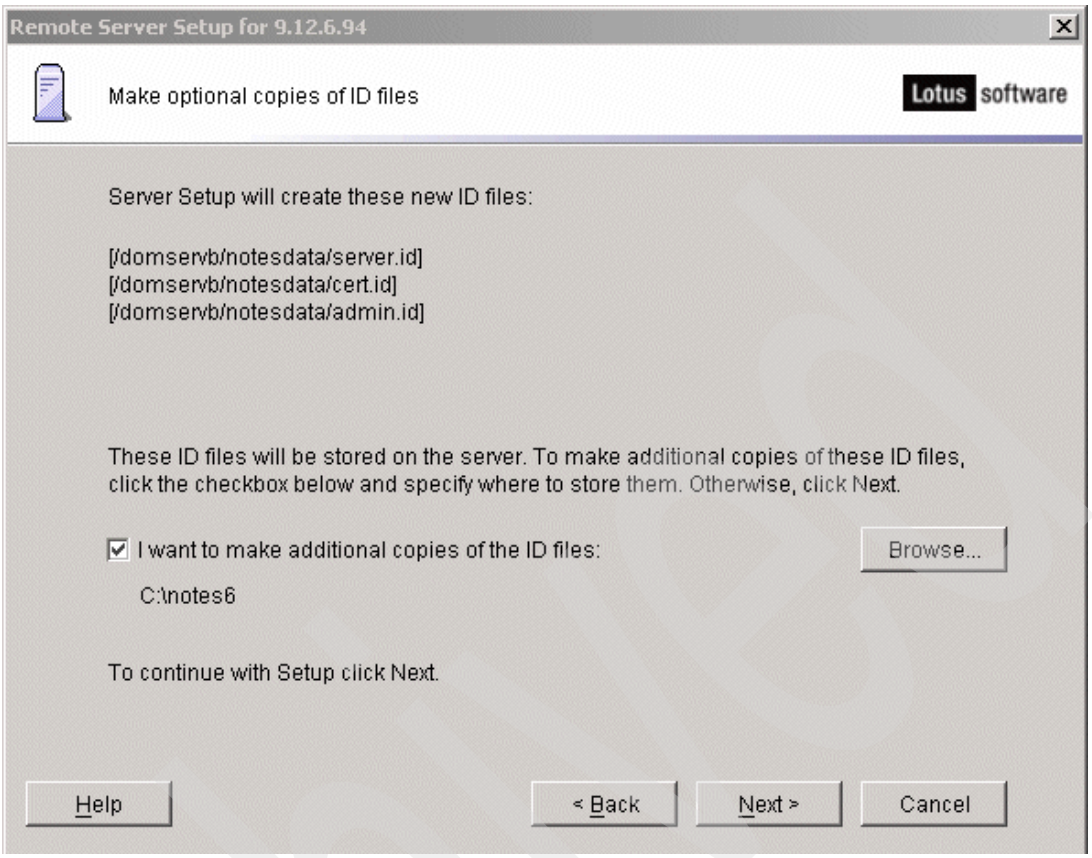



*Figure 7-31 Create a local copy of ID files*

31. The remote setup allows the server and certifier ID files to be copied to the local workstation. Make sure the check box is clicked. Remember the location for your ID files and keep them on a safe place as they are key to your Domino security.

    The screen shown in Figure 7-32 on page 164 gives you the opportunity to rethink your decision.

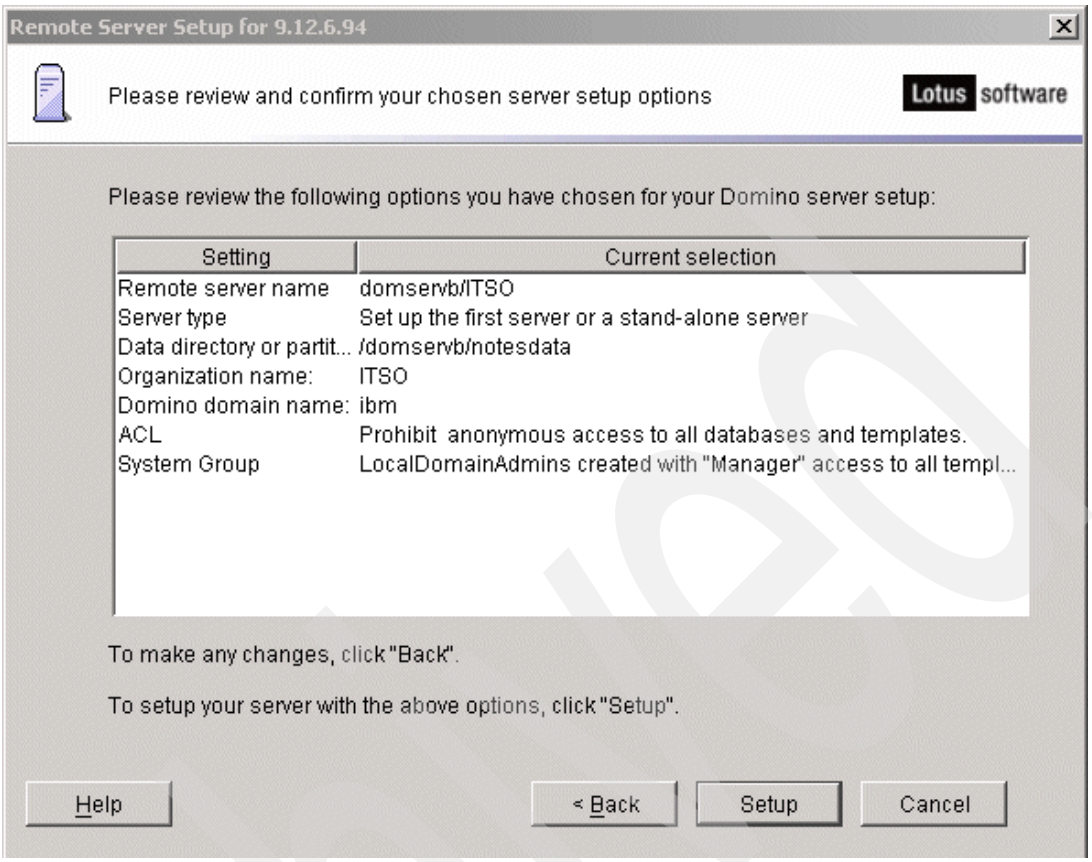



*Figure 7-32 Summary of the installation settings*

32. Look through the information and make sure everything is OK. If you need to make a change, press Back to go back. When you are satisfied that the information is correct, click **Setup** to finish the process. You can follow the progress, as shown in Figure 7-33.

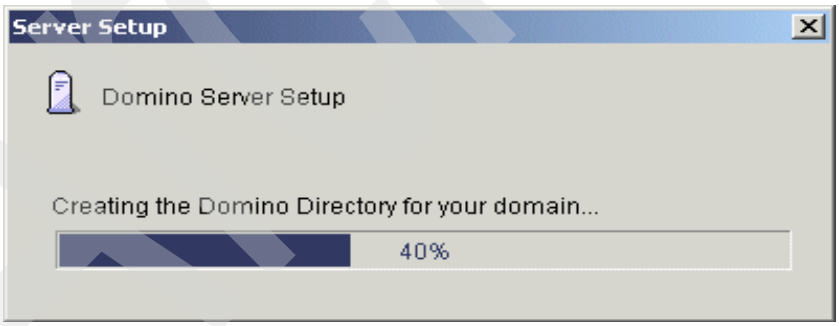



*Figure 7-33 Installation progress*

Once the installation is complete, you should see Figure 7-34 on page 165.

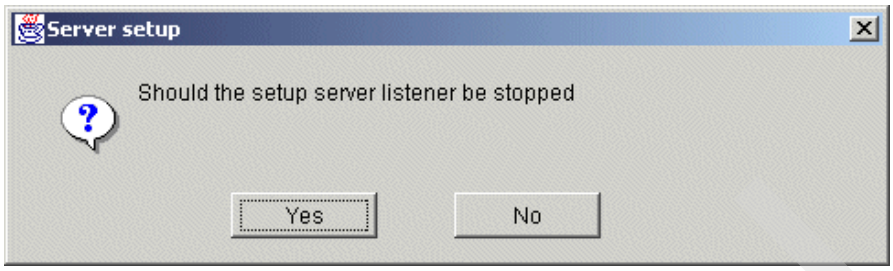



*Figure 7-34 Stop server listener task*

Click **Yes** to stop the Domino server command in the listen mode (see Figure 7-34).

33. After you stop the listener, no message will be displayed on the console. You are now ready to start your Domino server,

## 7.6 Starting the Domino server

To start the Domino server:

1. Log onto the system with the Domino user ID.
2. Ensure that you are in the notesdata directory.
3. Start the server with the command: **`/opt/lotus/bin/server`**.

This starts the server and you see the Domino startup messages on your screen.

For future startups, you might want to start with server with the **`-jc -c`** option; see Figure 7-35 on page 166. However the Domino console messages will not come to your screen.

To see these messages, you can log onto the server using the Domino Console; see 8.1.1, “Domino Controller and Domino Console” on page 174 for more information. If you want to have the messages come to your user console, start the server without the last **`-c`**.

```
domservb@linuxb:/domservb/notesdata # /opt/lotus/bin/server -jc -c
/proc/3019/mapped_base has been set to the recommended value of 134217728
The value in /proc/sys/fs/file-max is 26214, which does not match the recommended value
of 131072
The value in /proc/sys/net/ipv4/tcp_fin_timeout is 60, which does not match the
recommended value of 15
The value in /proc/sys/net/ipv4/tcp_max_syn_backlog is 1024, which does not match the
recommended value of 16384
The value in /proc/sys/net/ipv4/tcp_tw_reuse is 0, which does not match the recommended
value of 1
The value in /proc/sys/net/ipv4/ip_local_port_range is 32768    61000, which does not
match the recommended value of 1024        65535

Note:
If the Domino Server requires a password, you must first connect
to the Domino Server Controller via the Domino Console and enter the password to start
the Domino Server.


Domino Server Controller started at 8/15/03 5:29 PM.
Host name is linuxb.itso.com/10.32.16.24
Listening for connect requests on TCP Port:2050

Domino Server started at 8/15/03 5:29 PM.
```

*Figure 7-35 Startup of the Domino server*

If the Domino Server requires a password, you must first connect to the Domino Server Controller using the Domino Console, then enter the password to start the Domino Server.

In the installation guide, there is a section for optional post-installation steps with information about how to do the following:

- Create an additional organization certifier ID
- Create an organizational unit certifier ID
- Use Internet site documents to configure Internet protocol server tasks
- Enable the Internet sites view
- Create an Internet site document
- Set up security for Internet site documents

## 7.7 Installing the server executables using Script mode

Previously, we discussed the interactive mode of installation, and now we cover the script mode. This provides silent install functionality and allows you to install saved installation settings to a local server or remote servers.

Script.dat, the default sample script file, contains information you need to install the Domino server program files, including descriptions of each parameter and instructions for using the -script option to install partitioned servers. The script.dat file is located at /opt/lotus/zlinux directory.

Before changing the file, make a copy of the script file, for example, into myscript.dat. Then alter and add the information in myscript.dat to meet the requirements of your environment.

The script file is well documented, so spend some time getting familiar with its content. If you have done a interactive install before, it will be easier to understand all the options.

A number (#) sign indicates that it is a comment. Remove the # character if you want to use the setting. You find an example of the script.dat file in Appendix D, “Install script.dat” on page 385. For a typical installation, use the values provided in Table 7-4.

*Table 7-4 Recommended settings for the installation script:*

| Setting type | Recommended setting |
|---|---|
| Domino server installation type | Domino Enterprise Server : 2 |
| Install template files | template_install_option = 1 |
| Install server code | add_data_directories_only = 0 |
| Install ASP server | asp_install_option = 0 |
| Program directory | /opt/lotus |
| Create /opt/lotus soft link | opt_lotus_softlink = 0 |
| Data directory | / domserv#/notesdata |
| UNIX User name | domserv# |
| UNIX Group name | notes |

The list below helps you run steps for a script-based install:

1. Log in to the root account from your local system.
2. Copy the file /opt/lotus/zlinux/script.dat to /opt/lotus/zlinux/myscript.dat
3. Edit the file /opt/lotus/zlinux/myscript.dat to meet your requirements.
4. To install using the local script file, enter this command at the UNIX console prompt:

```
/opt/lotus/zlinux/install  -script /opt/lotus/zlinux/myscript.dat
```

## 7.8 Adding a partitioned Domino server (DPAR)

This section describes the process if you want to add an additional Domino server using the same set of executables to your Linux. This is known as Domino partitioning (DPAR). If you partition the server, multiple instances of Domino will share *one* set of program files, but each installation will have a separate data directory.

**Note:** The new Domino 6 feature that allows multiple installs cannot use DPARs because it requires separate program files, as well as separate data directories, for every Domino server instance. It thus requires more disk space than partitioning does, but adds flexibility.

We list here the steps necessary to add Domino partitions using the install script described in 7.7, “Installing the server executables using Script mode” on page 166:

1. Decide whether the new Domino servers will belong to an existing Domino domain.
   - If so, register the new Domino servers in that domain and save the server ID files on your local workstation. Also edit the server document in the Domino Directory to meet your requirements for security, ports, and servertask settings.

     Make sure you have a way to copy it to the Linux system, such as FTP.
   - If not, proceed with step 2.

2. Ensure that a Linux user ID exists for all new Domino servers. Do they belong to the correct group?
3. Ensure that you have defined a dedicated TCP/IP address for every Domino server on your Linux.
4. Create and mount the notesdata directory for each new server. Make sure they belong to the Domino user ID and that the permission bits are set correctly.
5. Update your script.dat file with add_data_directories_only=1. This will prevent the program directory from being written.
6. Set “installation_type”.
7. Specify the original program directory.
8. Only specify new server's notesdata at data_directories.

**Important:** Do not specify any server's data directory that was already installed.

9. Run the install script.
10. Copy the server ID files from step 1 on page 167 into the appropriate notesdata directory.
    a. If your new Domino server belongs to a separate Domino domain, you have to run the server setup as described in 7.5, “Setting up the Domino server” on page 152 and create the new certifier and server ID.

**Tip:** We do not recommend that you save server ID files in the Domino Directory. This is primarily because of security issues, but also because a server.id attached to the Domino Directory must be password-protected and this password will be required each time you start the server.

11. Logon with the Linux user ID that belongs to the Domino server and start the server.
12. Continue at 7.5, “Setting up the Domino server” on page 152 and run server setup for each Domino partition you want to install.

    If your new Domino servers belong to the same Domino domain as your old server, be sure to select “Set up an additional server” on the screen shown in Figure 7-23 on page 155.

## 7.9 Installing multiple Domino versions

You may decide to install multiple Domino servers within your Linux. Domino 6 now allows multiple servers with different levels of executing code to run within the same operating system. A new Domino version will have its own set of executable and its own notesdata directory. This is not the same as running a Domino partition (DPAR), where you have multiple Domino data directories using the *same* executable.

To run different levels of Domino code requires that the executables will be installed at different directory locations. For example:

1. /opt/D65/<version1>/lotus—contains Domino 6.5.x code
2. /opt/D65/<version2>/lotus—contains Domino 6.5.y code
3. /opt/D65/<version3>/lotus—contains Domino 6.0.2 code

For each of these versions, you can have multiple DPARs that share the same executable code. Figure 7-36 shows five Domino servers belonging to three different Domino domains within one Linux server.

More than one domain can be defined, where a domain encompasses the servers in its Domino Directory. This has nothing to do with the physical location of the code.

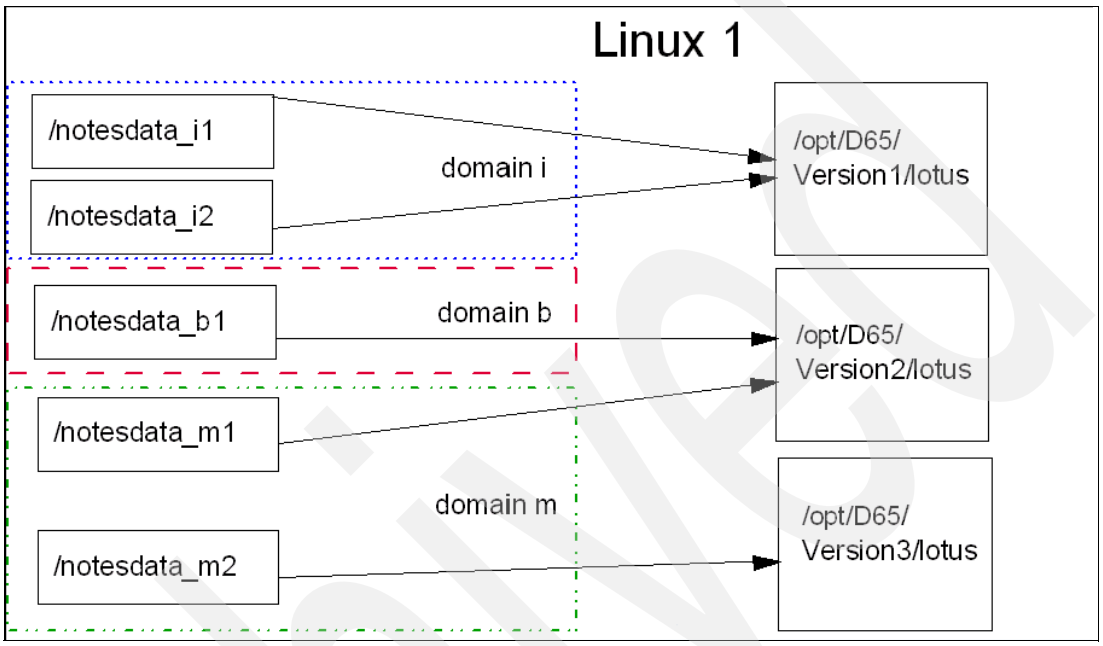



*Figure 7-36 Partitioned Domino servers*

If you choose to install the server executables in a different location than the default /opt/lotus as we did in step 9 on page 145, you get the option to put in a soft link for handling multiple Domino server installations. Domino server software no longer requires the /opt/lotus soft link for handling multi-Domino Server installations on a single operating system. You can now select your own option for the /opt/lotus soft link feature if you have chosen a directory other than /opt/lotus for installation; see Figure 7-37.

```
================================================================================
                          Domino Server Installation
================================================================================
    The Domino Server software no longer requires the /opt/lotus soft link for
    handling multi-Domino Server installations on a single operating system.
    You can now select your own option for the /opt/lotus soft link feature
    if you have chosen a directory other than /opt/lotus for installation.

--------------------------------------------------------------------------------
    Type e to exit the Install program.
    Press ESC to return to the previous screen.
    Press the Spacebar to change the setting until you get the one you want.
    Press TAB to accept a setting and continue to the next screen.
--------------------------------------------------------------------------------

>>>Do you want to create /opt/lotus soft link for this Domino Server? [No]
```

*Figure 7-37 Domino soft link option*

## 7.10 Avoiding installation problems

In this section we mention some of the installation problems that can occur.

### Cleaning up if setup fails

If your setup fails for some reason, or if you want to set up the server again, execute the following steps from notesdata directory to avoid problems when re-running the Domino server setup:

- Copy the saved skeleton notes.ini.orig to notes.ini.
- Remove names.nsf (if any).
- Remove admin4.nsf (if any).
- Remove setupdomwizard.nsf (if any).
- Remove the server ID and certifier ID created during previous setup session

### Do not install Domino on a non-supported Linux

If Domino is installed on a version of Linux on zSeries that does not support epoll I/O readiness facility, the following error message will be displayed and Notes client sessions will not be able to connect to the server.

```
This Linux kernel does not have the required epoll I/O readiness notification facility
Listener task exited:  Failed to create an IOCP port
```

See 2.6, “Linux planning and customizing” on page 20 for how to check whether epoll is installed.

### Invoking install by using the current script.dat file

The script.dat file supplies values used by the install code (incremental versus full installs).The install code checks for the correct version. Make sure you are using the version of the script.dat file that came with your current Domino version.

### Other Linux processes

Linux for zSeries may automatically start processes that will interfere with the Domino Server:

- Apache Web server
- SMTP
- ...

Close down these process if you experience any problems.

### Have enough disk space

You could exceed your DASD space either during installation of the Domino server or as the data directories grow over time. Use your zSeries systems management and capacity planning teams to ensure that you do not run out of disk space.

### Do not start Domino as root

If you try to start the server as root, Domino will not let you; see Figure 7-38 on page 171.

```
linuxb:/ # cd domservb
linuxb:/domservb # cd notesdata
linuxb:/domservb/notesdata # /opt/lotus/bin/server -jc -c


Do not run Domino as root.

linuxb:/domservb/notesdata #
```

*Figure 7-38 Start of Domino server as root*

8

# Domino administration

In this chapter, we describe functions and features that the Domino Administrator has available for monitoring, controlling, and maintaining a Domino server.

## 8.1 Ways to administer the Domino Server

In this section, we describe the various ways you can administer the Domino Server using:

1. Domino Controller and Domino Console
2. Domino Administrator
3. Web Administrator

You can administer the Domino system remotely with the Domino Administrator, the Web Administrator, or the Domino Console.

### 8.1.1 Domino Controller and Domino Console

The Domino Controller (also referred to as Server Controller) and Domino Console (also referred to as the Java console) are two new features for administering the Domino server beginning with Domino 6. Both features work together to provide administrators with secure access to Domino 6 servers from any location. This is the preferred method to remotely control your server. The files needed to run both the Domino Controller and the Domino Console are provided with Domino and Notes.

It also provides the best level of security, since any number of Domino Consoles can connect to a Domino Controller, but the administrators' identities are securely authenticated by their Domino Internet name and password. There is a also second level of security: the connection between Domino Console and Domino Controller is Secure Sockets Layer (SSL) encrypted, so data cannot be captured (or at least, understood) during the time it is in transit.

*Domino Controller* is new Java code that can be used to start the Domino 6 server. It also intercepts the output from the server and writes it to any connected Domino Consoles. Optionally, the controller writes the output to log files on the server rather than to the server's standard output location (usually the screen).

*Domino Console* is a new Java-based graphical user interface (GUI) that can be used on an administrator's local machine (any platform) to connect to a Domino 6 server that is being run by a Domino Controller, even to a server that is not responding to Notes clients. From Domino Console, administrators can securely send commands to be executed either in the Domino server or natively, by the operating system.[1]

The Domino Console functions strictly as a server console. You can use it to connect to a Domino Controller to monitor and issue commands to the server. However, it does not include the full set of Domino administration features that are available with the Domino Administrator and the Web Administrator, nor can it be used to open and manage Notes databases. Domino Console can also be used as an alternate/secure way to connect to a system running Domino. But it does offer the ability to issue shell or controller commands; for more information, refer to “Sending controller and shell commands” on page 183.

In addition to the Domino Console, you can also use remote consoles in the Domino Administrator and Web Administrator to communicate with the Domino Server.

### 8.1.2 Domino Administrator

The Domino Administrator is the administration client for Domino. You can use it to perform most administration tasks (for example, to open and manage Notes databases), just as you

[1] January/February 2003, Volume 9, Issue 1 of The View “The Administrator's Guide to Domino Controller and Domino Console - Secure Access to Domino 6 Servers from Any Location” by Jon Champlin

would on any other platform. In 8.3, “Monitoring the Domino system” on page 186, we discuss using the Domino Administrator for monitoring.

We used the Domino Administrator to list the databases in the data directory (notesdata) for DomServA (Figure 8-1).

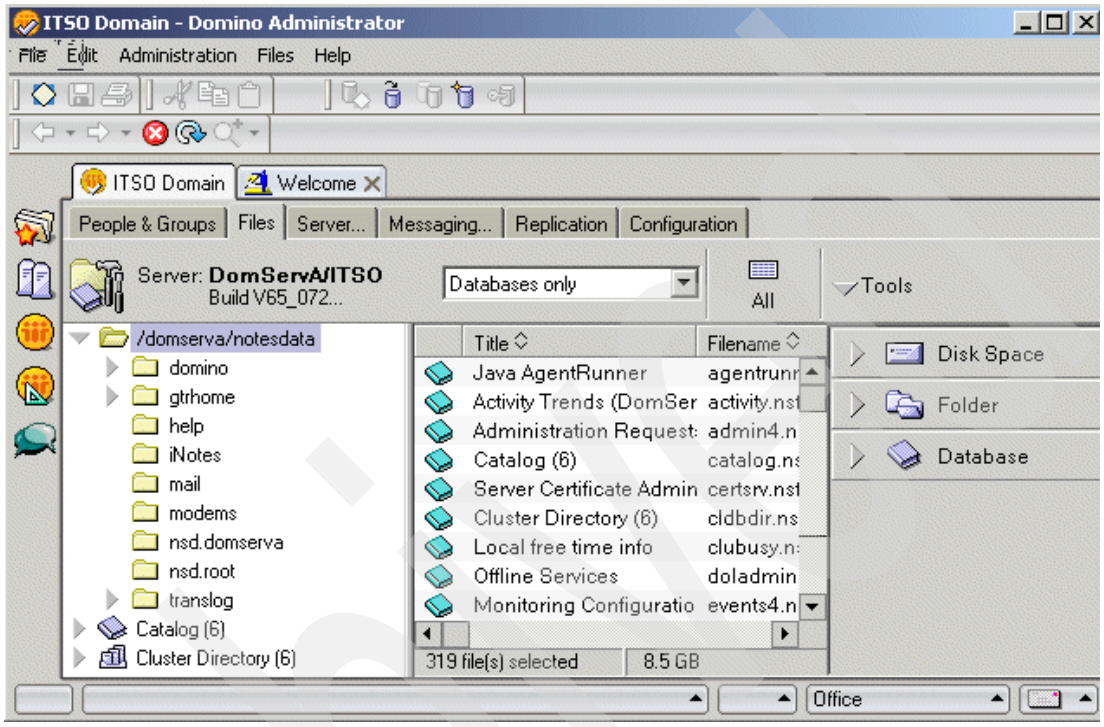



*Figure 8-1 Domino Administrator*

## 8.1.3 Web Administrator

You can start the Web Administrator using a browser, if you want to manage and view settings for a Domino server and perform most of the tasks that are available through the Domino Administrator. Once you are connected to the Web Administrator, it will indicate your level of access. Figure 8-2 on page 176 shows an Admin ID that has full access.

The Web Administrator uses the Web Administrator database (WEBADMIN.NSF). The first time the HTTP task starts on a Web server, Domino automatically creates this database in the Domino data directory.

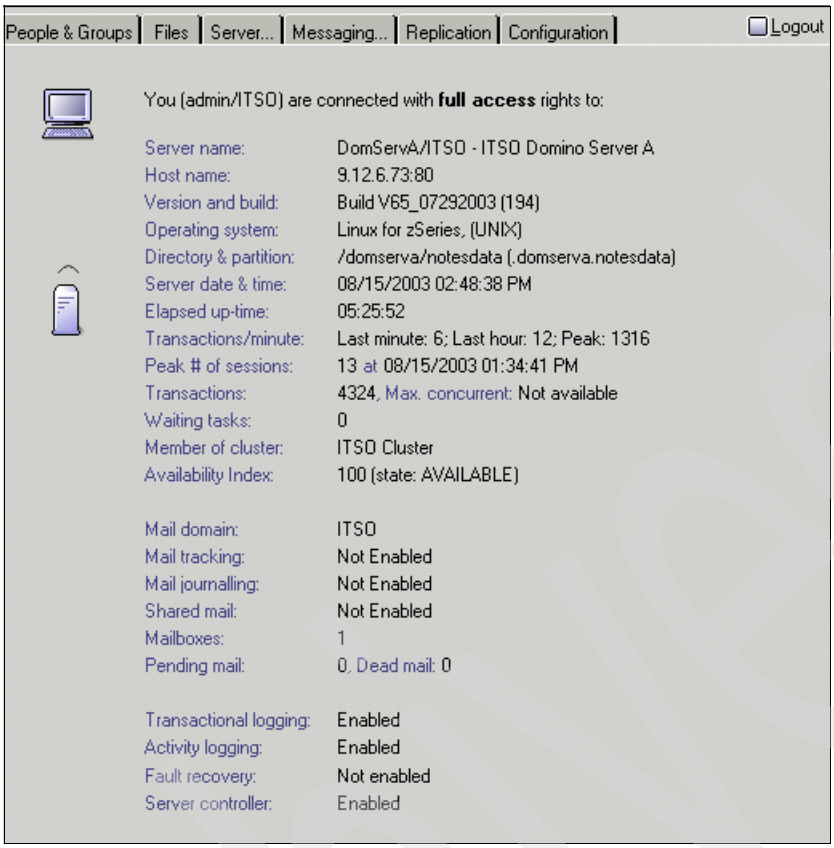



*Figure 8-2 Web Administrator*

To summarize, we have listed the various methods available to administer the Domino Server: Domino Console, Domino Administrator, and Web Administrator.

## 8.2 How to start and stop the server

Here we discuss the various ways you can start and stop the Domino Server. These choices are similar to those offered by the other platforms that Domino runs on, so most Domino Administrators will be familiar with the concepts and tools available.

In this section we will present the manual method of starting and stopping the Domino server using the Domino Controller and the Domino Console. However, for security reasons and ease of administration, we strongly recommend that you automate your server startup and shutdown procedure. Avoid using telnet or ssh to log into Linux and start a production Domino server. A sample automated startup and shutdown procedure is documented in 9.5, “Automating Domino startup and shutdown” on page 239.

### 8.2.1 Manually starting the Domino Controller and Domino Server

To manually start the Domino Controller, you first have to login to Linux with the Domino user ID. After verifying that you are in the notesdata directory, use the same command you use to start the Domino server but append the -jc parameter.

For example, **`/opt/lotus/bin/server -jc &`** will start the Domino Controller, the Domino Server (in the background), and the Domino Console.

**Tip:** Append an ampersand (&) at the end of the start command. Doing so will put the process in the background, so that it is safe to close the telnet or ssh session without terminating the Domino Controller or even the server (for example, `/opt/lotus/bin/server -jc &`).

When you run a Domino Controller, you no longer have access to the traditional console at the server. You can communicate only through the Domino Console or through a console in the Domino Administrator or Web Administrator.

As the Controller and Server are started, note that the kernel settings required by the Domino Server are correct. If these values were not tuned properly, the display will remind you of the recommendation and what is currently set.

**Reference:** The kernel settings are listed in Table 7-3 on page 137.

### 8.2.2 Display error when starting Domino Controller

In our testing, when we started the Domino Controller on Linux running as a guest on VM, we received errors that a DISPLAY variable was not set to allow the Domino Console to receive output, as shown in Example 8-1.

*Example 8-1 Starting Server using Domino Controller*

```
domservc@linuxc:/domservc/notesdata> /opt/lotus/bin/server -jc &
[1] 9394
domservc@linuxc:/domservc/notesdata> /proc/9394/mapped_base has been set to the recommended
value of 134217728
The value in / proc/sys/fs/file-max has been set to the recommended value of 131072
The value /proc/sys/net/ipv4/tcp_fin_timeout has been set to the recommended
value of 15
The value in /proc/sys/net/ipv4/tcp_max_syn_backlog has been set to the recommended
value of 16384
The value in /proc/sys/net/ipv4/tcp_tw_reuse has been set to the recommended value of 1
The value in /proc/sys/net/ipv4/ip_local_port_range has been set to the recommended
value of 1024   65535

The Domino Console cannot be started on zOS/zSeries:  No native GUI available.
Use the DISPLAY environment variable to redirect the Domino Console.
The Domino Console will not start unless the DISPLAY environment variable is set.

Domino Server Controller started at 8/18/03 4:15 PM.
Host name is linuxc.itso.company.com/184.156.23.129
Listening for connect requests on TCP Port:2050

Domino Server started at 8/18/03 4:15 PM.
```

Since there is no native GUI on zSeries, we got the DISPLAY error, but it only affects the redirect of the Domino Console. The Domino Controller and Server have been started successfully.

The DISPLAY errors were eliminated when we installed the Xlib on Linux and started an Xterm session and redirected the DISPLAY variable to point to a machine running an Xsession. However, the Xlib package on United Linux 1.0 is not installed by default. So you will need to plan for additional disk space to install it. We also observed a slower response to commands issue on the Domino console when it was configured in this way.

There are two optional arguments, **-c** and **-s**, you can specify when you start the server.

| | |
|---|---|
| server -jc -c | Starts the Domino Controller and the server, but not the Domino Console |
| server -jc -s | Starts the Domino Controller and the Domino Console, but prevents the Domino server from starting |

You may be wondering under what scenarios these options may be used.

Use **-c** to prevent the Domino Console from running when you start the Domino Controller. You might prevent the Domino Console from running on a slow machine or a machine that is low on memory. If you use this argument and the Domino server ID requires a password, the Domino server starts without running its server tasks. To run the server tasks, you must connect to the Domino server from a Domino Console and specify the server password when prompted.

Use **-s** to prevent the server from running when you start the Domino Controller. Use this argument along with **-c** so that someone who is directly at the server can start only the Domino Controller, and then a remote administrator can start the server and specify a required server password remotely from a console.

"When a Domino 6 server starts, it creates a file called "admindata.xml" in the server's data directory; this file contains information about each of the server's administrators. It contains the access level that each administrator or administrative group has been granted, and also contains each administrator's Internet password ("hashed"). The access settings determine whether a particular administrator can issue Domino Controller and shell commands from the Domino Console. These access settings are derived from the Administrators' settings in the server document in the Domino Directory." [2]

### 8.2.3 Starting and stopping the Domino Console

To start the Domino Console:

1. Make sure a Domino server or the Domino Administrator client is installed on the machine.
2. Run the following command directly from the program directory, or from a directory path that points to the program directory: **jconsole**

Figure 8-3 on page 179 shows the Domino Console.

[2] The View article, op. cit.

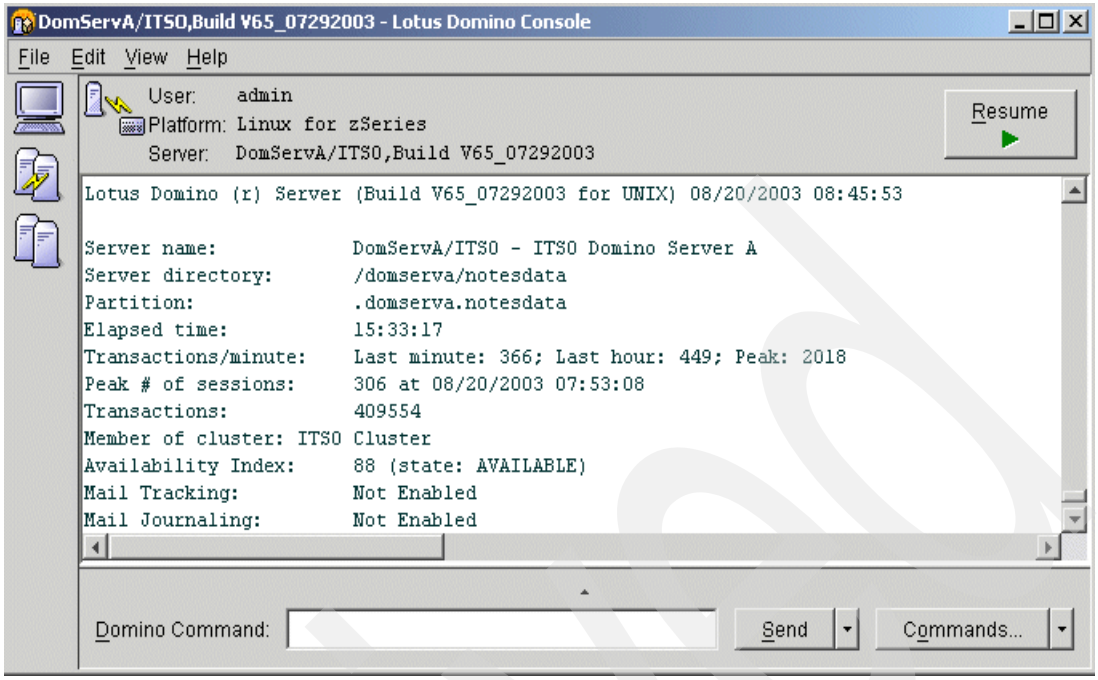



*Figure 8-3 Domino Console*

In order to use the Domino Console to communicate with a Domino server, the server must be running under a Domino Controller.

**Note:** If you need to start the Domino Console from a machine that does not have the Domino Administrator, you can copy the following files:

- jconsole.exe from the Domino program directory
- jvm directory and subdirectories
- dconsole.ini from where you ran the jconsole executable

To exit from the Domino Console, go to the Domino Console menu and select **File** → **Exit**, as shown in Figure 8-4 on page 180.

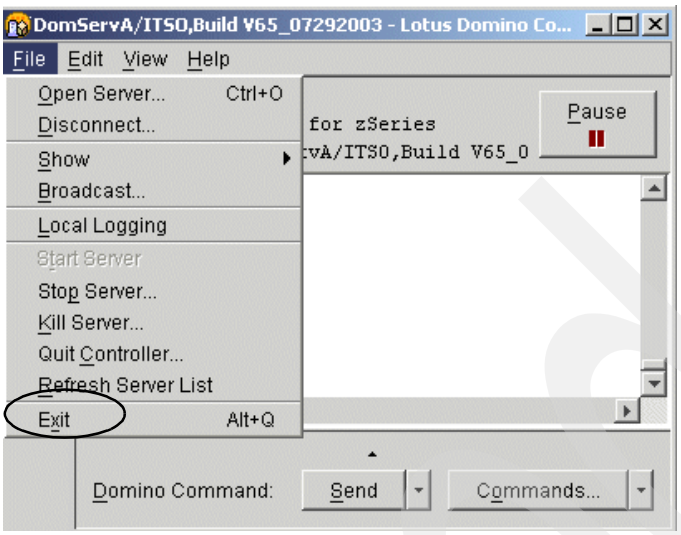



*Figure 8-4 Exit from Domino Console*

**Tip:** When the Domino Server is not responding, the Administrator Client and the Web Admin client cannot work, since they communicate with the server using NRPC and HTTP respectively. But the Domino Console can still talk to the Controller to find out what went wrong or tell the Controller to kill and restart Domino.

## 8.2.4 Stopping and starting the server from the Domino Console

There are times when an Administrator will want to shut down a Domino server. For example, to apply maintenance, update Domino server parameters that require a server restart, upgrade Domino code, run backups, and so on.

### Stopping the server

The way to stop the server gracefully from the Domino Console is to select from the menu **File** → **Stop Server**, as shown in Figure 8-5 on page 181.

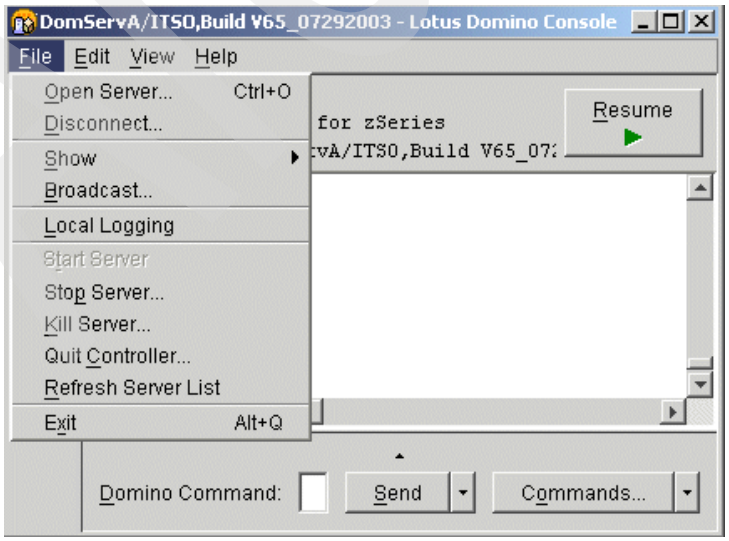


*Figure 8-5 Stopping server from Domino Console*

**Important:** It is not advisable to issue Kill Server for an orderly shutdown of the Domino server. When we tested this option, it did not do a clean shutdown of the server. We had to issue **`nsd -kill`** to remove shared memory that remained.

### Starting the server

When you are ready to start Domino again, select **File** → **Start Server**.

## 8.2.5 Logging server output

When you start the server using the Domino Controller, by default it automatically logs server output to log files on the server with the file name dcntrlr<creationdate>.log (for example, dcntrlr08141345.log). These log files contain the normal information you see at a server console.

By default, the Controller creates and logs server output to a new log file every seven days. You can change the number of days you want to keep your log with the Set ControllerLogExpiration=days setting.

If your log files are getting very big and you want to control the size of your logs, you should reduce the number of days. This change will take effect when you restart the Domino Controller or at midnight.

You can also change the name and specify a different path for your log files. However, you cannot customize the creation date or the extension portions of log file names.

These can be set using the Controller command **`# set setting`**, where setting is the name and value of the log setting or by updating these settings in the dcontroller.ini file. The various settings available are shown in Table 8-1.

*Table 8-1 Settings for Domino Controller*

| | |
|---|---|
| Set ControllerLogExpiration=days | Specifies the number of days' worth of log files to keep on the server. The default is seven days. |
| Set ControllerLogFileName=path filename | Specifies the name and path of log files created on a server. By default, log files are stored in the server's data directory with filenames that begin with the text dcntrlr, followed by the creation date, a sequence number and the file extension.log or.meta.<br><br>You can specify a different path, and can specify text to replace the dcntrlr portion of the log file names. |
| Set ControllerLogType=value | Specifies which type(s) of log file(s) to create on a server, or prevents the creation of log files.<br>► 0 -- Do not create log files<br>► 1 -- Create .log files that log only data normally seen at a console<br><br>Setting takes effect immediately |

On the server you will find the dcontroller.ini file (see Example 8-2) in the server's data directory. You can edit these settings directly in this file. Most changes will take effect at midnight, or when you restart the Domino Controller.

*Example 8-2 sample dcontroller.ini*

```
linuxa:/domserva # cat dcontroller.ini
[DController]
Server_Name=DomServA/ITSO
Server_Title=ITSO Domino Server A
Server_Domain=ITSO
Server_OSName=Linux for zSeries
Host_Name=linuxa
Port_Number=2050
ControllerLogFileName=dcntrlr
ControllerLogType=1
ControllerLogExpiration=7
CleanupScript=server -kn
AdminDataXmlReadFrequency=300
```

**Important:** The **#Set Controller <setting>** command issued from the Domino Console took effect only after our Domino Server was shut down.

## 8.2.6 Other uses of Domino Console

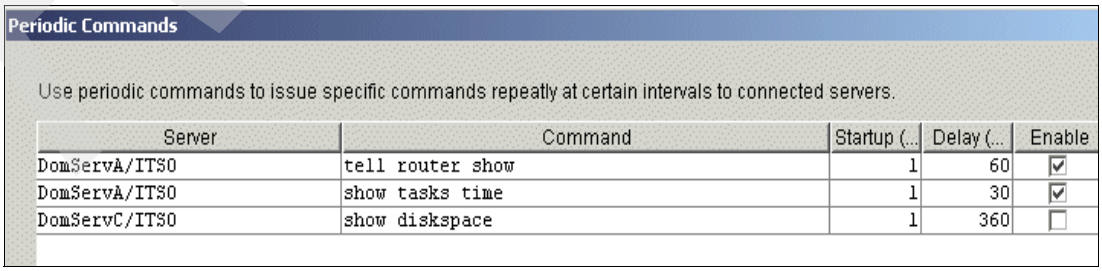
In addition to starting and stopping the server, the Domino Console has other uses, as described in this section.

### Periodic commands

You can use the Domino Console to send one or more commands to a server or group of servers at specified intervals by using the **Edit** -> **Periodic Commands** option. This provides the capability to quickly and easily set up one or several commands on your servers.

For example, we wanted to check the status of certain tasks, such as AMgr and Indexer, every 10 minutes. We created a periodic command to issue **show tasks time** on DomServA at ten-minute intervals, as shown in Figure 8-6. It also shows setting and enabling periodic commands on various servers.

- BelowStartup(m) specifies how soon (in minutes) after you enable the command that it first runs on the server.
- Below Delay(m) specifies the interval (in minutes) at which the command should run.




| Server | Command | Startup (... | Delay (... | Enable |
|---|---|---|---|---|
| DomServA/ITSO | tell router show | 1 | 60 | ☑ |
| DomServA/ITSO | show tasks time | 1 | 30 | ☑ |
| DomServC/ITSO | show diskspace | 1 | 360 | ☐ |

*Figure 8-6 Setting Periodic commands in the Domino Console*

## Sending controller and shell commands

You can also use the Domino console to send commands to the Controller or to issue shell commands.

### *Controller commands*

To send a **controller command**, use the prefix **#**.

For example, Figure 8-7 shows the tasks currently running on the Domino Server after issuing the controller command `#show Processes.`

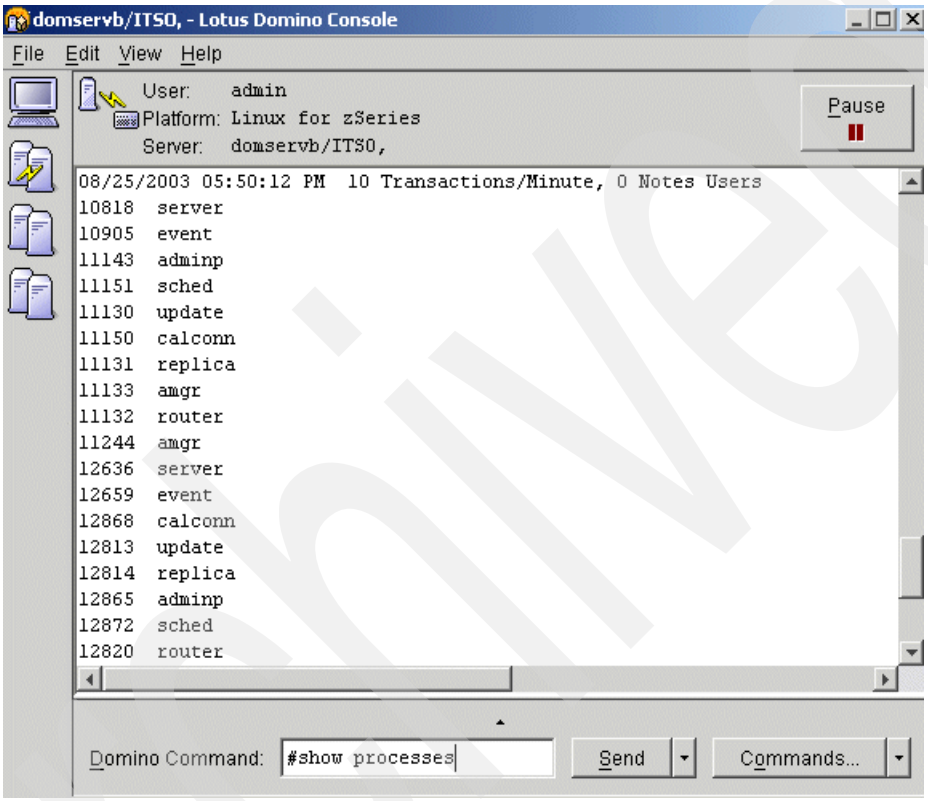



*Figure 8-7 The show processes controller command from the Domino Console*

For most commands, there is a also a Domino Console menu option that is the equivalent of the controller command. For example, **File - Show - Processes** is the menu option for the show processes controller command.

Table 8-2 shows some of the controller commands available. For a more complete listing, refer to the Domino Console Help function.

*Table 8-2 Example of Domino controller commands*

| Controller command | Description | Equivalent menu command in the Domino Console |
|---|---|---|
| Broadcast message | Broadcasts a specified message to all administrators connected to a Controller. | File - Broadcast |
| Kill Domino | Stops the processes on a Domino server that is not responding | File - Kill Server |

| Controller command | Description | Equivalent menu command in the Domino Console |
|---|---|---|
| Quit | Stops the Domino server and the server's Controller | File - Quit Controller |
| Start Domino | Starts the Domino server if it is down | File - Start Server |

### *Shell commands*

You can also send shell commands to a controller from a remote Domino Console. To send a **shell command,** use the prefix **$** to distinguish it from a Domino server command. For example:

```
$ ls *.log
```

Shell commands are executed in the server's data directory under the operating system permissions of the administrator who started the Domino server.

**Important:** The commands you can send to a server using a remote console are permitted by the level of administrator access you have in the Server document.

For more information about server events, see the *Domino Administrator 6 Help* database.

### Logging output locally

By default, the Domino server output is logged to files on the machine where the Domino server is running. If you want to inspect those files locally on your own machine without having to transfer them from the server, you can do so using the Domino Console.

Select the server, then select **Edit** -> **Local Logging** to specify the local logging parameters. After you have specified the PATH and file name and type of file you want to create (sequential or circular), then you must select **File** -> **Local Logging** to enable this feature.

## 8.2.7 Levels of access

In Domino 6, the access rights to the servers are defined differently. There are now seven different levels of access, ranging from full access to view-only capability. So you will need to plan carefully which access to assign to different users. The various access levels, and the Domino Controller or Domino Console commands they can issue, are listed in Table 8-3.

*Table 8-3 Levels of access*

| Access control field | Domino Server commands allowed | Domino Controller command allowed | Shell commands allowed |
|---|---|---|---|
| Full Access Administrators | All | All | All |
| Administrators/ Database Administrators | All | All | All |
| Full Remote Console Administrators | All | All | None |
| View-only Administrators | Commands that show status info only; for example, Show Tasks but not Load HTTP | All controller commands except Quit, Enable User, Disable User | None |

| Access control field | Domino Server commands allowed | Domino Controller command allowed | Shell commands allowed |
|---|---|---|---|
| System Administrators | None | All controller commands except Quit, Enable User, Disable User | All |
| Restricted System Administrators | None | All controller commands except Quit, Enable User, Disable User | Only shell commands listed in the Restricted System Commands field |

**Full Access Administrators** - new with Domino 6, and the highest level of administrative access to the server, gets all rights and privileges of all administration access levels listed. This level should be granted to only a few users, typically your most experienced administrators.

The full access administrator feature also replaces the need to run a Notes client locally on a server. It resolves access control problems—for example, those caused when the only managers of a database ACL have left an organization.

**Administrators** - the same access level used in Domino 5 and previous releases.

**Database Administrators** - responsible for administering databases on the server.

**Full Remote Console Administrators** - use the remote console to issue commands to the server.

**View-only Administrators** - can use the remote console to issue only those commands that provide system status information, such as SHOW TASKS and SHOW SERVER. However, at this access level, one cannot issue commands that affect the server's operation. As an example, there may a junior administrator who would be allowed to “view-only” to monitor the system and issue only a subset of commands.

**System Administrators** - allowed to issue a full range of operating system commands to the server.

**Restricted System Administrators** - allowed to issue only the operating system commands that are listed in the Restricted System Commands field shown in Figure 8-8 on page 186. For instance, commands such as `ls` or `vi -R` can be allowed to view files, but do not allow “rm” to actually delete files.

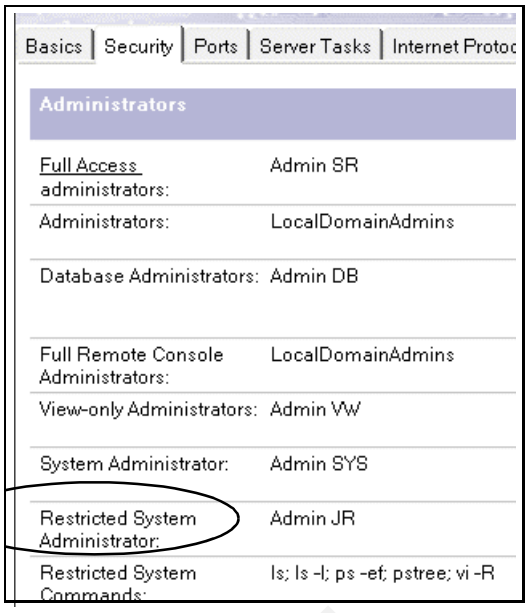



*Figure 8-8 Restricted system commands*

Therefore, depending on the needs in your organization, an administrator can assign the privileges required by individuals accessing the system. Each access level is fully explained in *Lotus Domino 6 Administration Help* database.

**Caution:** Administrators who are listed in the Full Access Administrators, Administrators, and Database Administrators fields on the Security tab of a server document are allowed to delete any database on that server, even if they are not listed as managers in the database ACL.

## 8.3 Monitoring the Domino system

Domino has many server monitoring features and tools that work together to inform you about the status, performance, and use of the Domino system. The Domino server provides services and tasks that create and report information about the Domino system in the form of statistics and events.

Domino generates statistics that you can use to monitor the activity of your servers and events. They indicate if the system is running smoothly, or signal you when there is a malfunction. You can monitor the availability and performance of your system using the Domino Administrator, the Domino Console, or the Web Administrator.

### 8.3.1 System monitoring tools

The Domino Administrator includes system-monitoring tools that you use to configure, view, and track the Domino system:

**Monitoring databases** - store monitoring documents, information, and results.

- The Monitoring Configuration database (EVENTS4.NSF) stores the documents you can use to set up monitoring. It also includes information about statistics, statistic thresholds, and event messages.

EVENTS4.NSF is used to configure Notes server event handling, statistic monitoring, ACL monitoring, and replication monitoring. It contains the names of all statistics monitored by the server, and thresholds for producing event records. It also contains error and status messages from the server. This database can be used to:

- Configure Notes server event handling
- Look up information about a specific statistic or event message
- Set statistic thresholds
- Assign types and severities to server events

- The Monitoring Results database (STATREP.NSF) stores the gathered statistics reports and can be configured to store information about logged events. The Statistic Collector task gathers statistics for one or more servers. It loads automatically on a server if it is listed in the task line (`ServerTasks=Replica,Router,AMgr,AdminP,...`) of the notes.ini file.

There are two ways to set up statistics collection:

1. Start the Statistic Collector task on each server, to collects its own statistics and create the report in the local Monitoring Results database
2. Start the Statistic Collector on one server that you set up to collect statistics from one or more servers and create reports in a specified “centralized” Monitoring Results database

**Monitoring Configuration documents** - define and configure what constitutes an event and how the event is handled.

You can also customize the messages that appear on the console when an event occurs. To configure an event, you need to determine the type of event, the severity level, and how you want it handled. You configure your events using the Event Generator and Event Handler documents.

- Event generators describe the condition that must be met for an event to be generated. Event generators gather information by monitoring a task or a statistic, or by probing a server for access or connectivity.

  Each event generator has a specified threshold or condition which, when met, causes an event to be created. The event is passed to the Event Monitor task, which checks whether an event handler has been defined. The Domino Administrator includes a set of default event generators, which are listed in the Event Generators view of the Monitoring Configuration database (EVENTS4.NSF).

- Event handlers describe what happens when the event occurs; for example, you can mail a notification of the event to a file or an administrator, or you can log the event to the log file (LOG.NSF). If you want to know about an event, you must have an Event Handler document; otherwise, the event is not recorded.

The Monitoring Configuration database (EVENTS4.NSF) includes default event handlers for server tasks. However, you can easily disable a default event handler and replace it with a customized one. To manually create an Event Handler document in the Monitoring Configuration database (EVENTS4.NSF), do the following:

1. From the Domino Administrator, click the **Configuration** tab, and open the Monitoring Configuration view.
2. Open the Event Handlers - All view, and click **New Event Handler**.
3. On the **Basics** tab, in the Server to monitor field, choose which server.
4. Under the Notification trigger, select one (that is, Any event that matches a criteria).
5. Complete the fields on the Event tab: event type, event severity, and message text
6. Click the **Action** tab, then select the notification method and select the enablement option.

**Note:** The Event Monitor task (formerly known as the Event task) starts automatically when you start the server, and it must run on all servers that you want to monitor.

**Server tasks** - collect and record information about the Domino system. The Event Monitor task determines if an Event Handler has been configured for the event, and if so, routes the event to the specified person, database, or server-management program for processing.

The Statistic collector task gathers Domino server statistics and creates statistics reports in the Monitoring Results database (STATREP.NSF) or to another database you can specify. The ISpy task executes TCP server and mail-routing event generators. However, we recommend that you use Ispy sparingly, since it does have a performance impact.

## 8.3.2 Server availability

You can check and monitor the availability of your Domino servers by using the Domino Console, Domino Administrator, or the Web Administrator. In addition, the Domino server monitor will provide you a way to display real-time statistics, as well as a visual representation of the status of servers and server tasks that you are monitoring.

You can also set up an event generator or *probe* to check on the availability of your servers on a regular basis. If you are currently using any third-party tools, you can check if they are also available for Domino on Linux.

### Domino Console

In the Domino Console, you can filter events and processes to check the status and availability of your server. You can filter server messages according to their severity. You can specify which server events the Domino Console displays in the console window.

By default, the Domino Console shows all server events. To allow filtering of the event types that you want displayed, expand the bottom panel of Domino Console by clicking the up/down arrow that is directly above the Domino Command field; or from the View menu, select **Event Filter**.

If you want to display only error messages, then in the Command panel, select only the type of server events you want to show in the Domino console window (see Figure 8-9 on page 189). For example:

- Fatal
- Failure
- Warning (High)

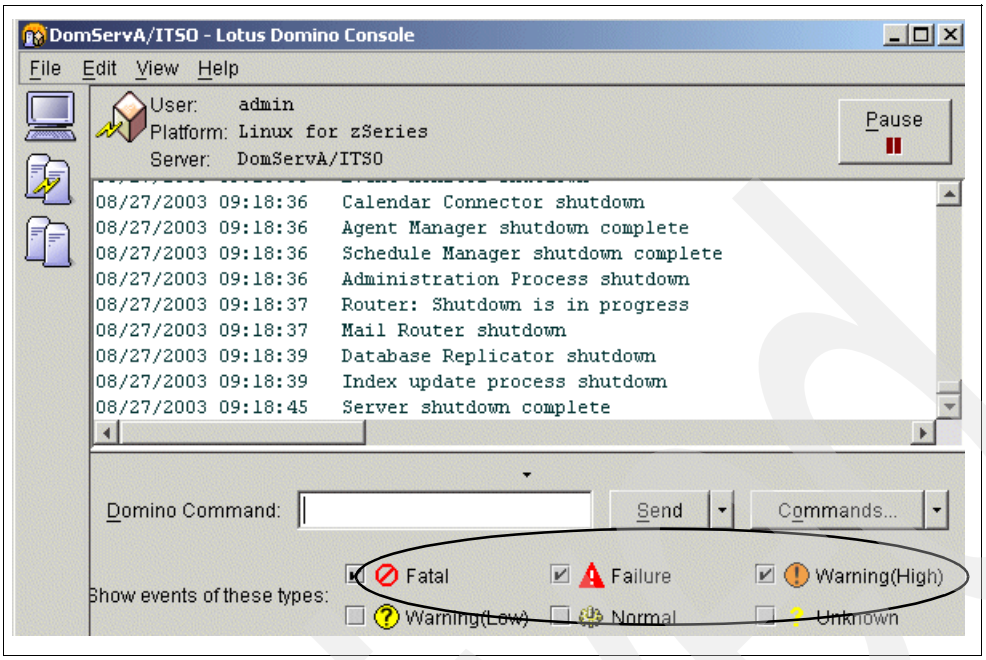



*Figure 8-9 Domino Console filter messages*

For more information about server events, refer to the *Domino Administrator 6 Help* database.

## Domino Server Monitor

In the Domino Administrator, with the Domino Server Monitor, you can view all your servers or just a subset of servers. Some of the tasks you can perform with the server monitor is to:

- View server monitor statistics by timeline or by state
- Display past error states
- Add or remove a server to monitor
- Add or remove server tasks or statistics from a selected server or from all servers

**Note:** The Domino server monitor is not available in the Web Administrator.

Start and stop the Domino server monitor manually, as follows:

1. From the Domino Administrator, click the **Server - Monitoring** tab.
2. Click the Green arrow in the upper-right of the task screen.
3. To stop the server monitor, click **Stop**.

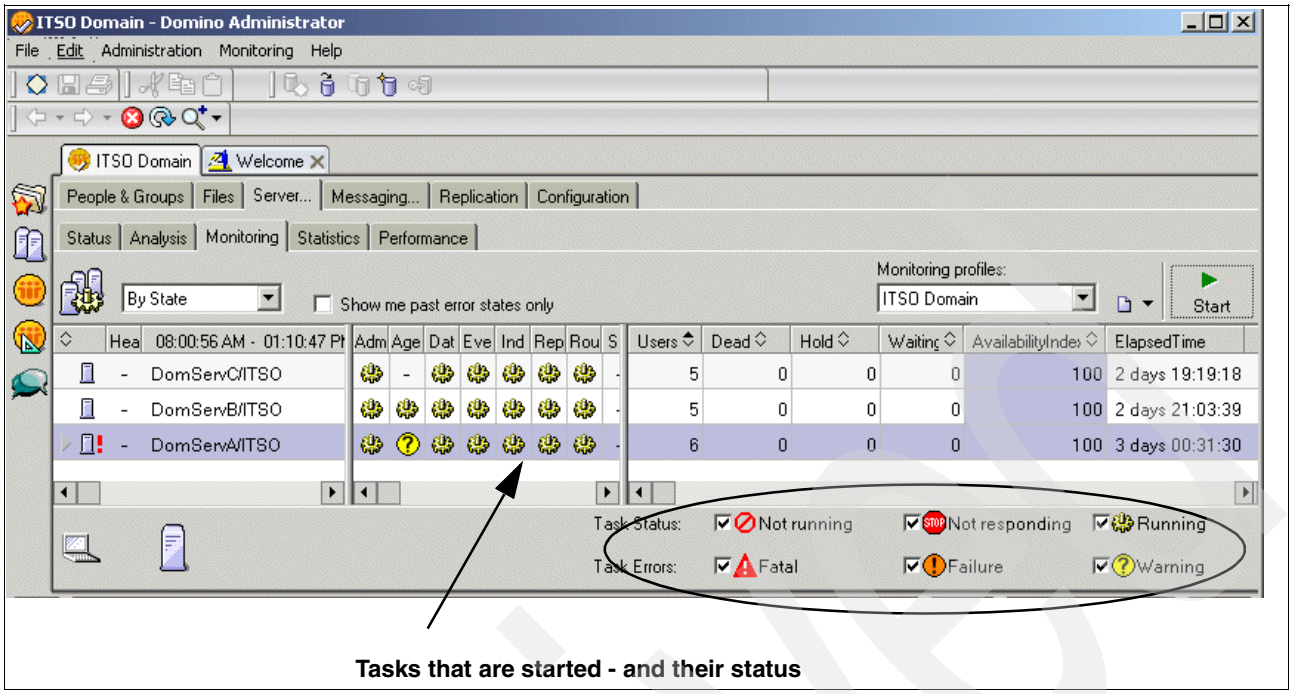



*Figure 8-10 Domino Server monitor*

Each server and server task displays a status indicator that identifies its current state: running, not running, or not responding, as shown in Figure 8-10. You can use the option "Display past states reporting errors exclusively" to only view error states

.

> **Note:** The Domino server monitor does not start by default. Change the monitoring defaults in the Administration Preferences to start it automatically.

## Domino Server event generator

Another way to check for the availability of your Domino Server is to create a Domino Server event generator. There are numerous server probes or events you can define; you have to decide which ones are useful in your environment.

You can set up your "probe" or server event as follows:

1. From the Domino Administrator, click the **Configuration** tab, and then open the Monitoring Configuration view.
2. Open the Event Generators - Domino Server Response view, and then click **New Domino Server Event Generator**.

We defined two kind of probes:

1. To open a database to check if the server is running and alive, as shown in Figure 8-11 on page 191
2. Sending mail to check if the router task is working, as shown in Figure 8-12 on page 192

Of course, both probes also check the network connection between the servers.

We set up a "probe" server event that checked the connectivity and port status of the two servers, DomServA/ITSO and DomServC/ITSO, by opening the readme.nsf file every six minutes (the default is every three minutes). You will not want to select a large file, since it will take a long time to open.

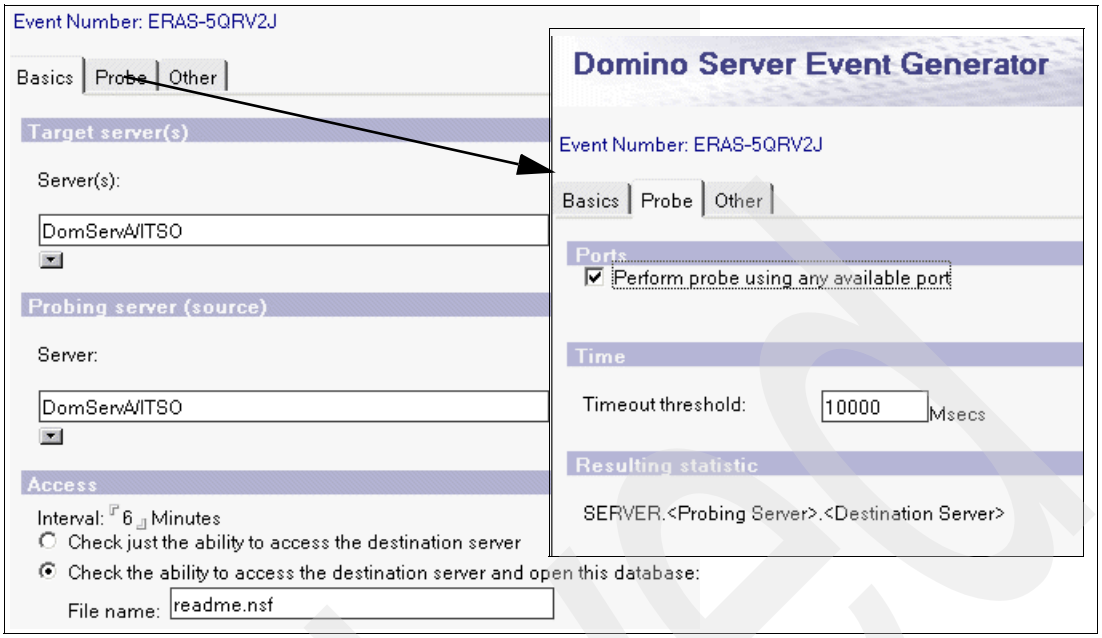



*Figure 8-11 Server Availability probe*

Set up the mail probe as follows:

1. Make sure the ISpy task is running.
2. From the Domino Administrator, click the **Configuration** tab, and then open the Monitoring Configuration view.
3. Open the Event Generators -> Mail view, and click **New Mail Routing Event Generator**.
4. On the Basics tab, complete the fields listed in Table 8-4.

*Table 8-4 Mail probe - basic settings*

| Field | Action |
|---|---|
| All Domino servers in the domain will probe themselves | Do one:<br>- Check this option to have each server to probe only the local mail box.<br>- Uncheck this option to probe specified servers. |
| Recipient | Enter the address of the recipient for which you want to check the mail route or use the drop-down box to select a recipient from a Domino Directory or Address Book. Do not enter more than one user and do not enter a group name. |
| Probing servers (source) | Select the name of the server from which to start the probe. |
| Show intermediate hop times | Enable this option to track intermediate hop times. |

5. Click the Probe tab and complete the fields shown in Table 8-5.

*Table 8-5 Mail probe - probe settings*

| Field | Action |
|---|---|
| Send interval | Enter the number of minutes between probes. The default is 15. |
| Time-out threshold | Enter the number of minutes the probing server (source) waits for a response before logging a failure. |

6.Click the **Other** tab, complete fields shown in Table 8-6, and then click **Save and Close**.

*Table 8-6 Mail probe - miscellaneous settings*

| Field | Action |
|---|---|
| On time-out, generate a Mail event of severity | Select the severity level. |
| Create a new event handler for this event | Click this button to launch the Event Notification Wizard and create an event handler. |

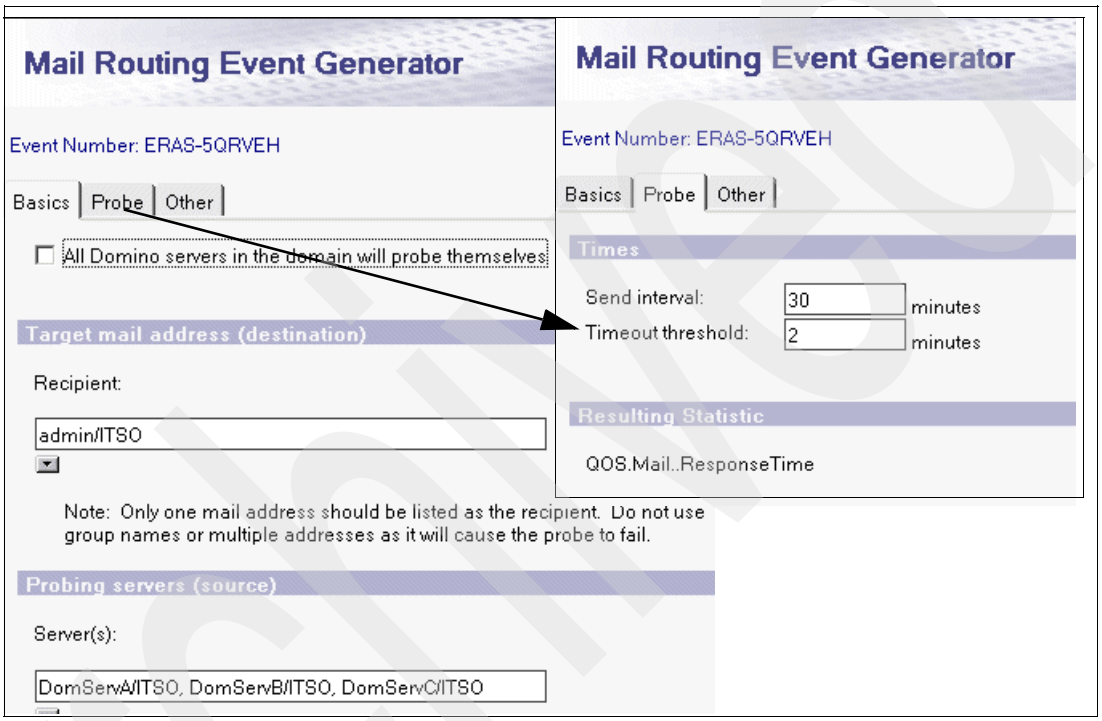



*Figure 8-12 Mail probe*

## 8.3.3 Performance monitoring

Domino generates statistics that you can use to monitor your system performance and utilization. You can use the Domino and platform statistics or Tivoli® Analyzer for Domino to determine the health of your servers.

When you run Linux under VM, remember that all (performance) data depends on the resources dedicated to Linux on any specific point in time. VM keeps changing the amount of CPU and memory. So each time Linux tries to find out what the percentage of allocated memory is, the absolute value may have changed. So any absolute value should be correct, but any relative (percentage) value may not be.

### Domino statistics and VM

When you run Domino under VM, be aware that all the system resources that Domino and Linux are measuring are virtual resources. So VM makes Linux think it has nnn MB of memory or xx CPUs. If those values were static, everything would be fine—but they are constantly changing. Typically, you define ranges for memory and CPU that VM can allocate to a guest. VM reallocates them if another guest has a higher demand for resources.

## Domino statistics

Domino gathers statistics that show the status of processes currently running on the system. For example, the statistic “Mail.TransferThreads.Active” indicates the number of active mail threads. This tells you something about the status of the router. You use these statistics, along with the predetermined statistics thresholds, to monitor both your Domino system and platform statistics.

In addition to platform statistics, you can display other Domino statistics by issuing **`Show Stat <statisticname>`**.You can also use **`load stats <servername>`** to create statistics on demand for a remote server, and **`show statistics server`** to display the complete list of statistics for the server. Here are some of the statistics that you can use for performance monitoring:

- **Server.Trans.PerMinute.Peak** is the peak number of transactions during a one-minute interval since the server started. It is not the peak during this interval. You are given the date and time this occurred.
- **Server.Users** is the number of users with connections to the server at the end of the interval when statistics were collected.
- **Server.Users.Active15Min** is the number of active users in a 15-minute interval.

There are also counters for the number of calendar and mail requests, and also agent processes. These may be of use if you are investigating the average profile of your mail users.

Just as you create an event generator for a Domino system statistic, you also create an event generator for a health statistic. Then, when the statistic does not meet the defined threshold, an event is generated.

For an event to be created, however, you must enable statistic alarms. Then, the first time a statistic alarm is reported, an event is generated and reported to the in STATREP.NSF. In addition to an alarm, you can create an event handler to notify you of the event.

To enable statistic alarms

1. From the Domino Administrator, choose **File - Preferences - Administration Preferences**.
2. Click **Statistics**, and then check: `Check statistic alarms while monitoring or charting statistics`.
3. For: `Check alarms every <n> minutes (greater than monitoring poll interval)`, enter a value that is greater than the server polling value. The default is 15.

## Coordinating Linux, VM, and Domino data collection

To get Domino to collect records at the same time as VM and Linux, you must start the statistics task at the desired time, since it does not run at a specific time, but only after a defined period of time as elapsed.

If you want to combine Domino statistics with Linux or VM data, for example, you will probably need to extract the Domino statistics into file. You can then use currently available tools to analyze the data.

For example, from the Domino Administrator, you can use the Domino Server monitor and statistics charts to view graphical representations of system status. From the Domino Console, you can view a representation that uses your predefined colors and text attributes to illustrate the status of a process.

## Processing Domino statistics

To process Domino statistics, you can:

- View them in the statrep database.
- View them using the Administrator Client or by value (Figure 8-13),under the Statistics tab.

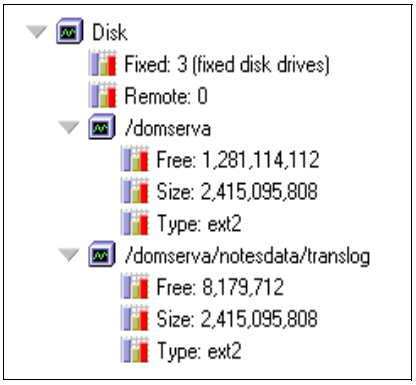



*Figure 8-13 Statistics for free and total Disk space*

- View them graphically as shown in Figure 8-14. You find this under the Performance tab of the Server tab of the administration client.

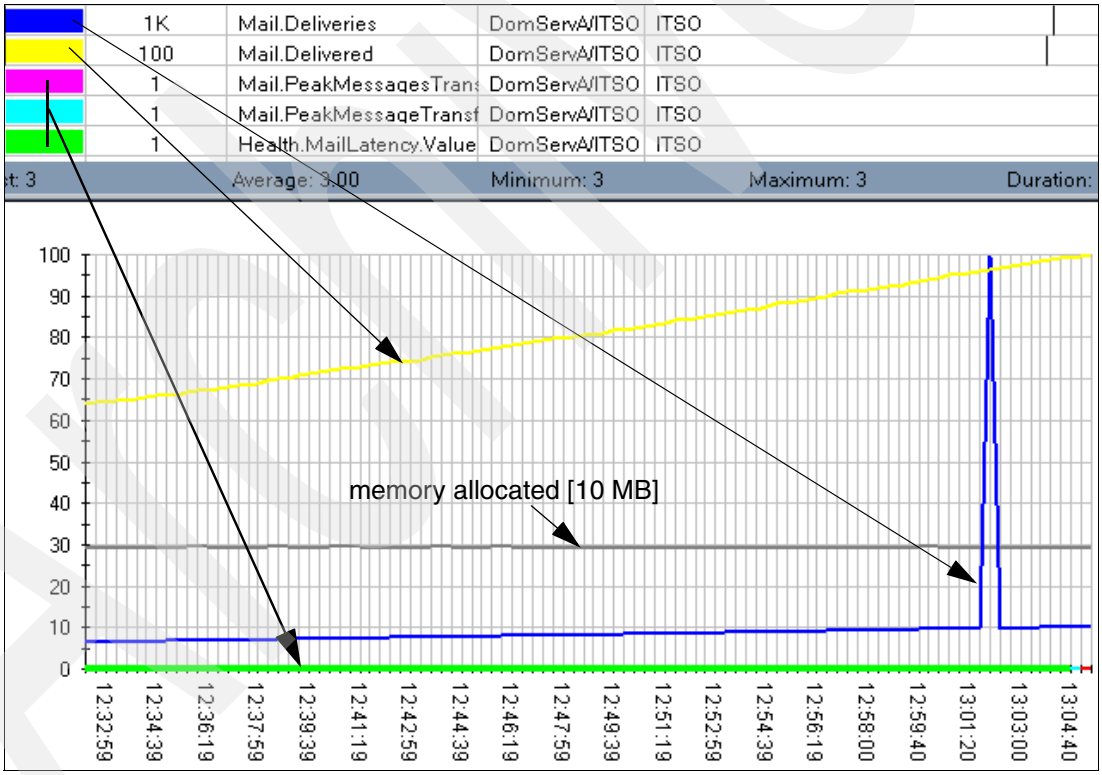



*Figure 8-14 Graphical representation - Domino Server statistics*

## Platform statistics

Platform statistics are now available for Linux on zSeries. They provide another way to gain insight into the combined behavior of Domino and Linux. Performance information is gathered from Linux and stored as Domino statistics that can be collected and processed just like any other Domino stats.

However, when you are running Domino on Linux under z/VM, data depends on the resources dedicated to Linux on any specific point in time, so the memory and system statistics will not reflect your configuration. If you are running Domino “natively”, the statistics will be correct.

Platform statistics are collected continuously by the Statistic Collector. You can view these statistics from the Domino Administrator in the Statistics tab, as shown in Figure 8-15.

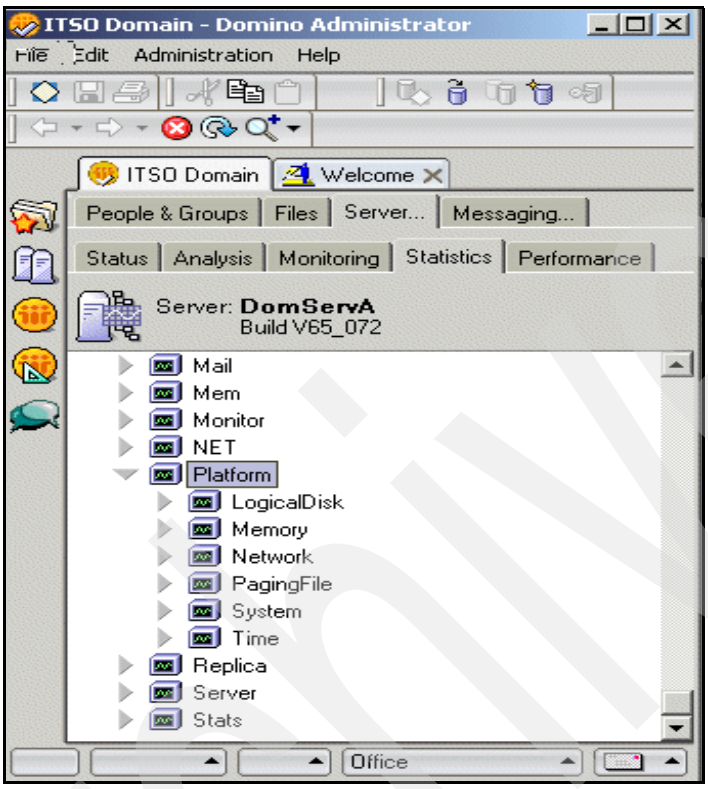



*Figure 8-15 Platform statistics*

The following are the statistics that are gathered:

**Logical disk** Statistics for individual disks and total percent use of all disks

**Memory** Statistics showing memory allocation and use, including available memory

**Network** Statistics for individual network adapters and cumulatively for all the network adapters on the system

**Paging file** Statistics that show use of paging files

**System** Statistics on the information captured—for example, a summary of system CPU use and queue length

**Time** The time that platform stats were last collected, and the sampling interval in minutes

No disk space is consumed by enabling platform statistics, since no log files are created. As with Domino statistics, disk space is used only if you log platform statistics to the log file or to the Monitoring Results database (STATREP.NSF). The amount of disk space used depends on the frequency of capture.

You can also use the **show stat platform** command from the Domino console to view all platform statistics, or just a subset.

- **Show Stat Platform** displays a complete list of platform statistics.
- **Show Stat platform.logicaldisk.*** displays all the platform statistics in the logical disk group.

Example 8-3 shows an extract of the results obtained when we issued the **Show Stat platform.logicaldisk.*** command from the Domino Console.

*Example 8-3 Logical disk*

```
Show Stat platform.logicaldisk.*
  Platform.LogicalDisk.10.AssignedName = dasdl
  Platform.LogicalDisk.10.AvgQueueLen = 0.1
  Platform.LogicalDisk.10.AvgQueueLen.Avg = 0.3
  Platform.LogicalDisk.10.AvgQueueLen.Peak = 2.49
  Platform.LogicalDisk.10.PctUtil = 0.7
  Platform.LogicalDisk.10.PctUtil.Avg = 1.92
  Platform.LogicalDisk.10.PctUtil.Peak = 11.43
  Platform.LogicalDisk.10.ServiceTime = 23.73
  Platform.LogicalDisk.10.ServiceTime.Avg = 31.97
  Platform.LogicalDisk.10.ServiceTime.Peak = 2,623.08
  Platform.LogicalDisk.11.AssignedName = dasdm
  Platform.LogicalDisk.11.AvgQueueLen = 0.1
  Platform.LogicalDisk.11.AvgQueueLen.Avg = 0.24
  Platform.LogicalDisk.11.AvgQueueLen.Peak = 2.95
  Platform.LogicalDisk.11.PctUtil = 0.68
  Platform.LogicalDisk.11.PctUtil.Avg = 1.79
  Platform.LogicalDisk.11.PctUtil.Peak = 10.43
  Platform.LogicalDisk.11.ServiceTime = 21.58
  Platform.LogicalDisk.11.ServiceTime.Avg = 32.48
  Platform.LogicalDisk.11.ServiceTime.Peak = 3,850
```

**Important:** If iostats is not installed on the system platform, stats will not display the disk stats. iostats is typically delivered in the sysstat package on Linux. Refer to 2.6.2, “Linux sysstat package” on page 21 for more information about this subject.

DomServA was running on a Linux guest under VM and had two CPUs and 256 memory. Example 8-4 is an extract of the results we received from **show stat platform**.

*Example 8-4 show stat platform display*

```
Memory
Platform.Memory.PagesPerSec = 0
  Platform.Memory.RAM.AvailMBytes = 3
  Platform.Memory.RAM.AvailMBytes.Avg = 2
  Platform.Memory.RAM.AvailMBytes.Min = 2
  Platform.Memory.RAM.AvailMBytes.Peak = 9
  Platform.Memory.RAM.PctUtil = 98
  Platform.Memory.RAM.TotalMBytes = 186

Paging
Platform.PagingFile.Free.SizeMBytes = 419
  Platform.PagingFile.Total.PctUtil = 13
  Platform.PagingFile.Total.PctUtil.Avg = 10
  Platform.PagingFile.Total.PctUtil.Peak = 13
  Platform.PagingFile.Total.SizeMBytes = 484
```

```
System
Platform.System.ContextSwitchesPerSec = 1,738
  Platform.System.ContextSwitchesPerSec.Avg = 1,712
  Platform.System.ContextSwitchesPerSec.Min = 1,479
  Platform.System.ContextSwitchesPerSec.Peak = 2,565
  Platform.System.PctCombinedCpuUtil = 2
  Platform.System.PctCombinedCpuUtil.Avg = 9.86
  Platform.System.PctCombinedCpuUtil.Peak = 45
  Platform.System.PctTotalPrivilegedCpuUtil = 1
  Platform.System.PctTotalPrivilegedCpuUtil.Avg = 4.5
  Platform.System.PctTotalPrivilegedCpuUtil.Peak = 21
  Platform.System.PctTotalUserCpuUtil = 1
  Platform.System.PctTotalUserCpuUtil.Avg = 5.35
  Platform.System.PctTotalUserCpuUtil.Peak = 24
Time
Platform.Time.LastSample = 08/26/2003 18:46:29 EDT
  Platform.Time.SampleRateInMins = 1
503 statistics found
```

**Note:** When collecting statistics from a partitioned server, Domino collects platform statistics that pertain to the system as a whole, not to an individual server.

## Statistics and reporting database

If you run the reporter task on the server, a set of reports will be created at scheduled intervals. These are put in the statistics and reporting database statrep.nsf. You can specify the interval between records. For information about setting up the reporter task, refer to the *Domino 6 Administration Help* database.

Domino 6 includes a number of improvements that make it easier for administrators to view server reporting information.

| Feature | Description | Benefit |
|---|---|---|
| Statistic Charting | Administrators can now see current and historical graphing of Domino and platform statistics from the Administrator client. | Administrators can assess server performance and behavior in a historical context and in real-time. |
| Database monitoring | Reorganized DB monitor form, single-click creation of new database usage, activity, replication, and ACL monitors. | Simplified access to monitoring. |
| Event Description | Right-click access to more detailed information about console messages. | Contextual access to information. |

## Activity logging

Activity logging can be used to collect information about the activity in your enterprise. This information can be used to charge users for the amount they use your system, monitor usage, conduct resource planning, and determine if clustering would improve the efficiency of your system.

Domino writes the activity logging information in the Domino log file (LOG.NSF). To create activity logging reports, you write a Notes API program to access the information in the log file. You can also view the activity logging information by using Activity Analysis.

In a hosted environment, enable activity logging on all of your ASP servers. These are the servers used to house and maintain your hosted organizations

You configure activity logging by editing the Configurations Settings document.

1. From the Domino Administrator, click the **Configuration** tab.
2. In the Task pane, expand Server and click **Configurations**.
3. In the Results pane, select the Configuration Settings document you want, and click **Edit Configuration**.
4. On the Configuration Settings document, click the **Activity Logging** tab.
5. Select **Activity logging is enabled**.
6. In the Enabled logging types field, select the types of activity you want to log.
7. (Optional) To increase or decrease the frequency of creating Checkpoint records, change the checkpoint interval.
8. (Optional) To automatically create Notes session and Notes database Checkpoint records every day at midnight, select: `Log checkpoint at midnight`.
9. (Optional) To automatically create Notes session and Notes database Checkpoint records every day at the beginning and end of a specific time period, select: `Log checkpoints for prime shift` and then specify the times for the Prime shift interval.
10. Click **Save and Close**.
11. (Optional) If you are logging activity for LDAP Add and Modify operations, and want to change the amount of information logged in the Attributes field from the default of 4096 bytes, follow the steps in the topic “Limiting the amount of attribute information logged for LDAP Add and LDAP Modify activity.”

## Tivoli Analyzer for Lotus Domino

IBM Tivoli Analyzer for Lotus Domino takes advantage of the new Lotus Domino 6 statistics and activity measurements, and runs seamlessly inside the Domino Administrator. It generates comprehensive, detailed statistics and measurements of the server's activity

Tivoli Analyzer for Lotus Domino includes two integrated system-management tools:

- Activity Trends, which provides data collection, data exploration, and resource balancing
- The Server Health Monitor, which offers real-time assessment and recommendations for server performance

We list and describe these tools in Table 8-7, “Tivoli Analyzer for Lotus Domino tools” on page 199.

*Table 8-7 Tivoli Analyzer for Lotus Domino tools*

| Feature | Description | Benefit |
|---|---|---|
| Activity Trends Reporting | This reporting capability lets administrators analyze server workload by user, database, and protocol.<br>They can also determine growth rate trends. | Administrators can use this reporting to predict life span of existing disk and processing resources. They can also identify the databases responsible for server activity.<br>In addition, Activity Trends Reporting automates creation and execution of workload redistribution and server decommissioning plans. |
| Server Health Monitoring | Health Monitoring tracks blends of Domino and OS statistics from within the Domino Administrator client to determine overall server health.<br>Incorporated drill-down technology allows an administrator to find specific counters (for example, CPU, disk queue) responsible for reduced performance, and to suggest both immediate and long-term corrective actions. | This tool helps administrators determine, at any time, just how well a server is performing.<br>It isolates trouble spots automatically and suggests corrective actions. |

Tivoli Analyzer for Lotus Domino is a separate product offering from Tivoli Systems. You can obtain more information about this tool at:

http://www.ibm.com/software/tivoli/r/analyzerfordomino

The Server Health Monitor is part of the IBM Tivoli Analyzer for Lotus Domino. It integrates with the Domino Server monitor, which is part of the Domino Administration client. You can use it for monitoring and troubleshooting the performance of your Domino server. It automatically calculates health statistics and compares those statistics to predefined thresholds. It also reports on the overall health of the server.

To set up the Server Health Monitor, complete these steps:

1. Install the IBM Tivoli Analyzer for Lotus Domino.
   a. Make sure you have installed the Domino Administrator.
   b. Run the install program (SETUP.EXE) from the Tivoli Analyzer directory.
2. Start the Domino server monitor.

To view server health:

1. Make sure you have enabled the Server Health Monitor in Administration Preferences, started the Domino Server Monitor, and allowed the monitor to run for a few minutes longer than the specified polling interval.
2. From the Domino Administrator, click the **Server - Monitoring** tab.
3. In the Health column (shown as “Hea” in Figure 8-16 on page 200), the Server Health Monitor uses these icons to indicate the server's overall health:

**Green thermometer** The server's overall health rating is Healthy. All server components are within the appropriate range.

**Yellow thermometer** The server's overall health rating is Warning. One or more server components being monitored are approaching unacceptably poor levels of performance.

**Red thermometer** The server's overall health rating is Critical. One or more server components being monitored are failing to perform within acceptable tolerance levels.

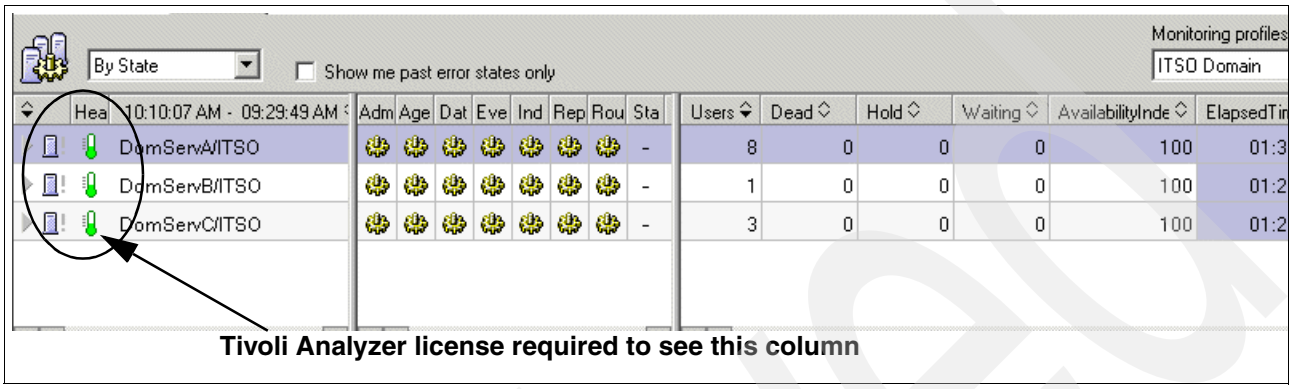



*Figure 8-16 View server health*

The Server Health Monitor reports a health rating for each server being monitored and for all enabled individual server components such as these; see Figure 8-17.

- CPU, disk, memory, and network utilization
- NRPC name lookup
- Mail delivery latency
- Server response

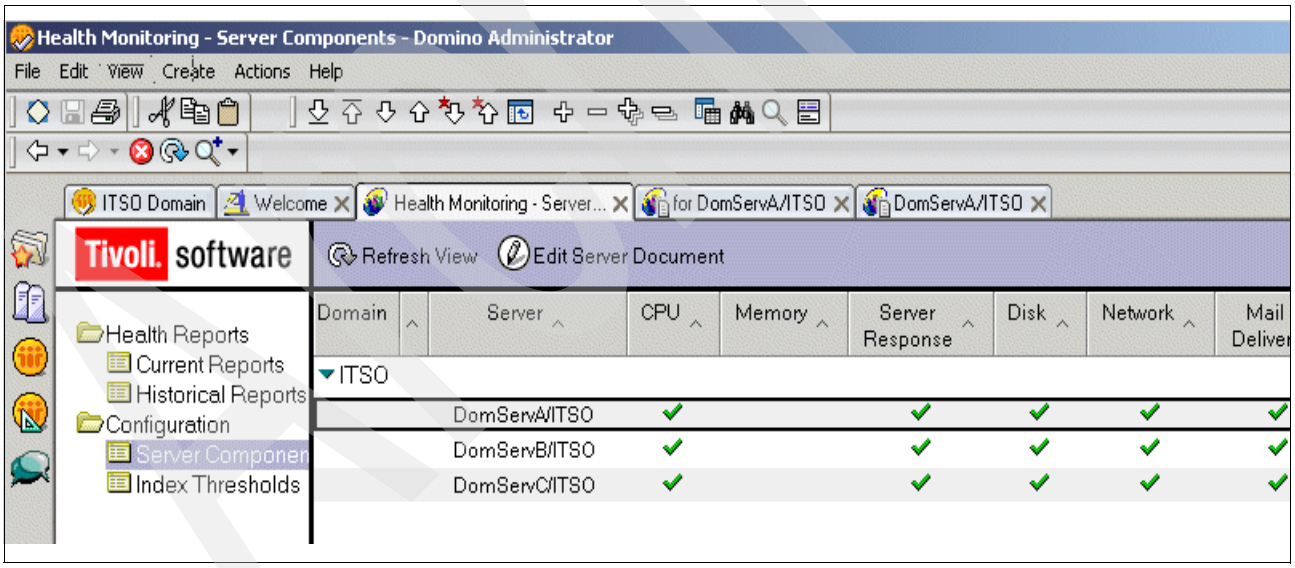



*Figure 8-17 Health monitoring*

If the server health rating is Warning or Critical, a health report is generated in the Health Monitoring database (DOMMON.NSF) locally on the Domino Administration client. The health report makes short-term and long-term recommendations for tuning the server and returning the performance status to a Healthy state.

To view the Health reports:

1. From the Domino Administrator, click the **Server - Monitoring** tab.

2. From the menu, choose **Monitoring - Switch to Health Reports**.

In the following health report (Figure 8-18), CPU, disk, and network utilization are displaying a critical rating, since the Disk Utilization component is reporting critically poor performance. The overall health report contains short- and long-term recommendations, as shown in Figure 8-19 on page 202.

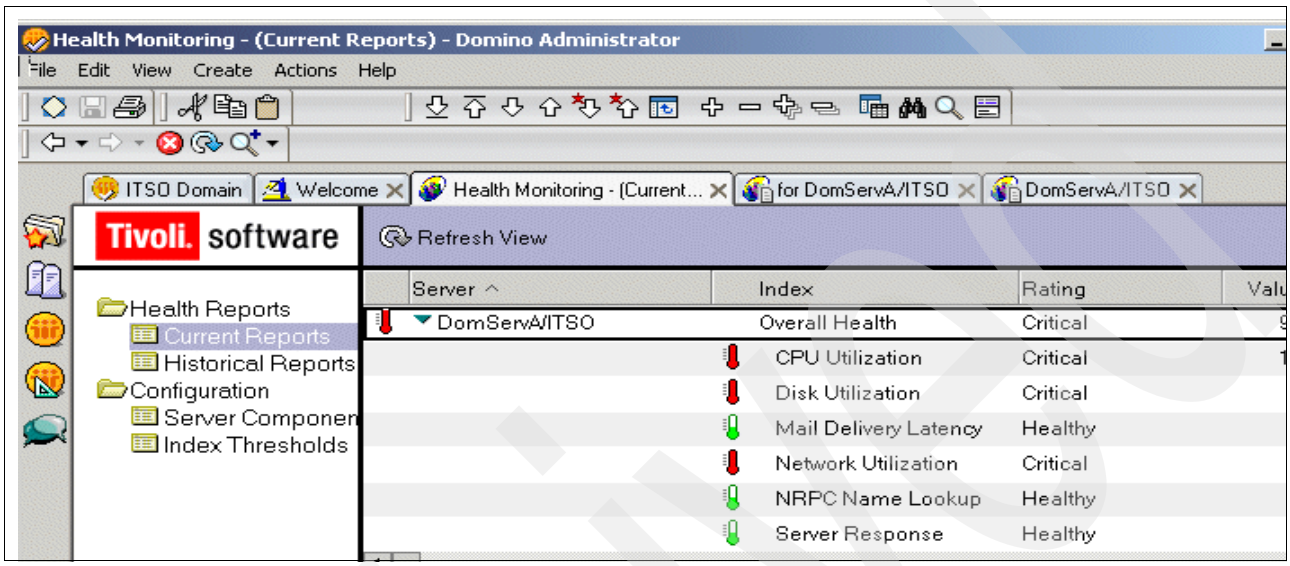



*Figure 8-18 Current Health Report*

Server Health Monitor does not include threshold values specific to Linux on zSeries, and it automatically picked up the thresholds for Linux/Intel which do not include CPU, disk, and memory. Although the CPU and disk utilization values were at a low value, this was reported as “critical” and we were not able to change this to a higher value. Therefore, at the time of the writing, you are not able to modify these specific thresholds; but this is expected to be made available. However, you can modify the other thresholds (for example, server response and mail delivery latency).

The Index Thresholds view in the Health Monitoring database (DOMMON.NSF) displays the threshold values for each platform.

To modify a threshold value:

1. From the Domino Administrator, click the **Server - Monitoring** tab.
2. From the menu, choose **Monitoring - Display Health Reports**.
3. Under Configuration, choose **Index Thresholds**.
4. Choose the operating system whose threshold you want to change, and choose **Edit Threshold Document**.
5. Change the value for the Warning Threshold or Critical Threshold.
6. Click **OK**.

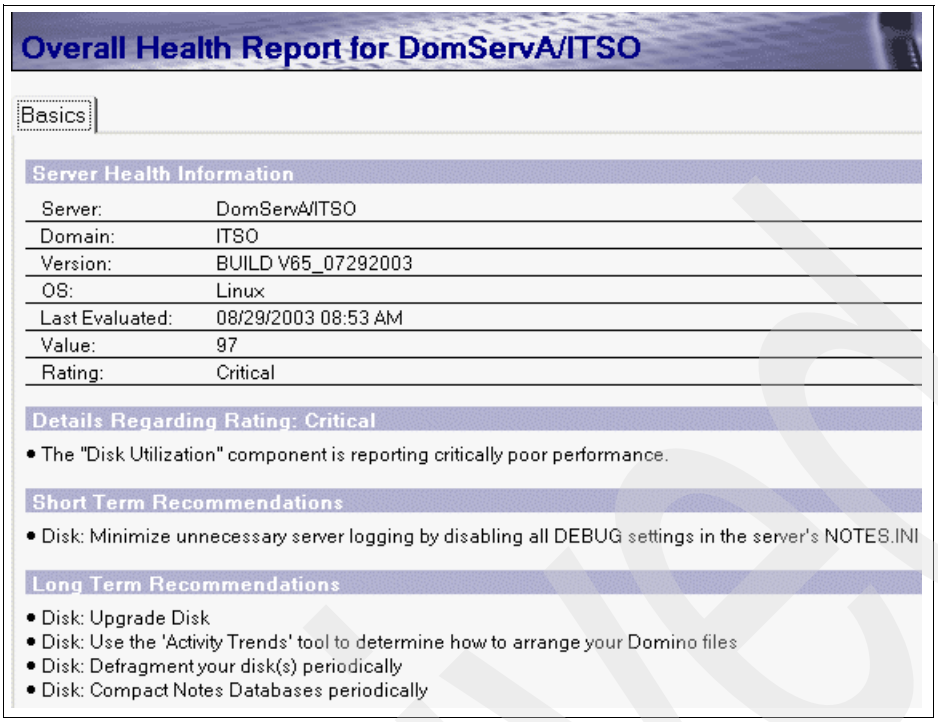



*Figure 8-19 Overall Health Report*

## 8.4 Transaction logging

A *transaction log* is a record of changes made to Notes databases. The transaction log consists of log extents and the log control file (NLOGCTRL.LFH).

A log extent is one of the log files into which the transaction logs are written. It has the form Sxxxxxxx.TXN, where x character represents a seven-digit number that is unique to that server. Domino fills each extent sequentially before writing data to a new one. Each server has only one transaction log, which captures all the changes to databases that are enabled for transaction logging.

Domino Release 6 supports transaction logging and recovery. With this feature enabled, the system captures database changes and writes them to the transaction log. Then, if a system or media failure occurs, or a user needs to have documents or a whole data base restored, you can use the transaction log and a backup utility to recover your database to either its most current state, or to any point in time.

All changes to a Domino database are made by transactions applied to the database using a defined set of procedure calls. There is no other way of updating the Domino databases. Domino transaction logging logs both the data and the actual transaction commands used to make the change; for example: “NSFNoteOpen, NoteUpdate, NoteClose.”

Transaction logging provides these main benefits:

- In most situations, you no longer need to run the Fixup task to recover databases following a system failure.
- Faster recovery after crash. When a server terminates abnormally, databases can be left in an inconsistent state. When transaction logging is not used, the server will scan all databases it can find at startup time to see if any are in this inconsistent state. Consistency checks, a type of fixup, are performed on these databases to remove the

inconsistencies. This operation typically requires a great deal of time for large installations that have many databases.

When transaction logging is employed, only those databases that have had updates in the log since the last transaction log checkpoint was written need to be processed at server startup. Since only the small specific number of updates need to be applied to this set of databases, this greatly reduces the processing time. There is no need to scan all of the documents in the database to see if they are all correct.

- Transaction logging saves processing time because it allows Domino to defer database updates to disk during periods of high server activity. Transactions are recorded sequentially in the log files, which is much quicker than database updates to random, nonsequential parts of a disk. Because the transactions are already recorded, Domino can safely defer database updates until a period of low server activity.
- More efficient handling of disk I/O. Transaction logging may improve performance and save I/O processing time because it allows Domino to defer database updates to disk during periods of high server activity. Transactions are committed sequentially to the logfiles, typically within about 1/100 of a second of the completion of a transaction, rather than being committed to the database on disk. This is much quicker than database updates to random, nonsequential parts of a disk. Since writes to disk are deferred, they can be consolidated, and thus more work can be done with fewer operations. Because the transactions are already recorded on disk in the transaction log, Domino can safely defer database updates until a period of low server activity.

The backup benefits of transaction logging are discussed in 9.7, “Backup/recovery” on page 249.

### 8.4.1 Setting up transaction logging

The following procedure shows you how to set up transaction logging:

1. Ensure that all databases to be logged reside in the Domino data directory, either at the root or in subdirectories.
2. From the Domino Administrator, click the **Configuration** tab, expand the Server section, and click **All Server Documents**.
3. Select the Server Document for the Domino server you want to edit and then click **Edit Server**.
4. Click **Server**, and then click **Current Server Document**.
5. Click the **Transactional Logging** tab.
6. Decide which logging style you will use; the choices here are Circular, Linear, or Archival. Circular (the default) is the easiest to begin with and includes a slight performance gain. Whichever method you choose, be sure that you have enough disk space.
7. Enable “Transactional logging.”; refer to Figure 8-20 on page 204.
8. Complete other fields as appropriate, and then save the document. See *Domino 6 Administration Help* database for more details.
9. Stop and restart the Domino server to have the changes take effect.

**Note:** The initial setup for transaction logging can result in a long period of unavailability while the server assigns DBIIDs to each database; it could take several hours.

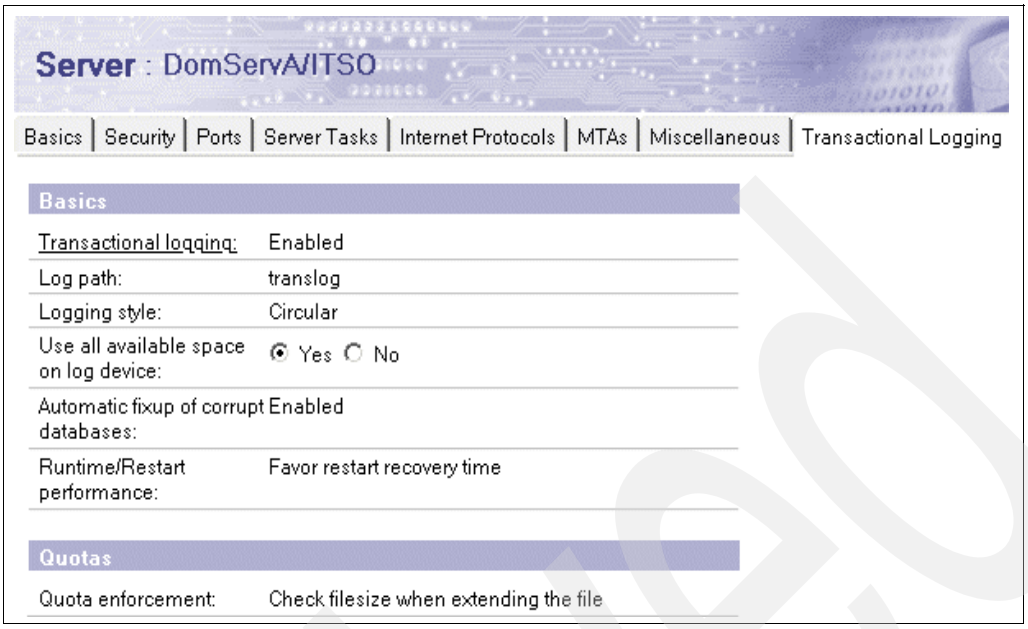



*Figure 8-20 Enabling transaction logging*

## 8.4.2 View logging

*View logging* is a feature in Domino 6 that allows transaction logging of view elements.When enabled, view logging logs complex view information to the transaction log. If a crash of a server occurs, the log and the view can be reloaded from the transaction log, eliminating the need for a view rebuild.

There is a slight cost associated with view logging, which causes increased CPU utilization when using this features, so it should only be enabled for complex views. This feature is off by default for most databases. It must be enabled for each database view using the Domino Designer® client.

### Enabling view logging

To enable view logging, use Domino Designer. In Designer, open a view or folder, select the **Advanced** tab, and check: `Logging - Include updates in transaction log.`; see Figure 8-21 on page 205.

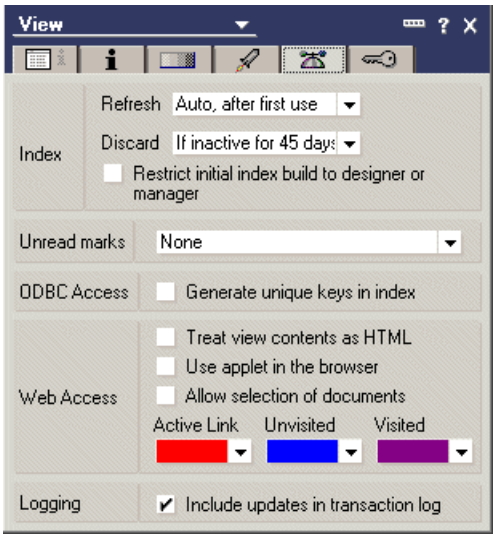



*Figure 8-21 Enabling view logging*

## Soft deletes

The process of restoring a database, in this case a mail file, can be time-consuming for the administrators on any platform. This is especially true when the restore is needed to recover a single document that was recently accidently deleted by the user.

The new design of the mail file standard template for Domino 6 includes a Trash view. Soft deletes are enabled by default. When a user deletes a document, it is placed in the Trash view and is recoverable by the user for 48 hours. This default time period can be changed; see Figure 8-22 on page 206.

The issue of quotas is addressed by specifying “Check space used in file when adding a note” in the quota enforcement field on the Transaction Logging tab (Figure 8-23 on page 206). When this option is specified and user deletes a document, the space occupied by that message is immediately removed from the calculated size of the mail file. There is no need to run the Compact task to recover space.

Users who cannot receive mail because of a quota violation can reduce the current size of the mail file immediately by archiving or deleting messages. Since the calculation of quota does not include the items in the trash (soft deletes), it is not necessary to compact the database until reaching the normal weekly database maintenance window. Users who need to undelete a document can do so without the necessity (and overhead) of having an administrator restore their mail file.

### *Changing the retention period for soft deletes*

The retention period of soft-deleted documents can be set by changing the value in the “Softdelete expire time in hours” value on the database properties tab shown in Figure 8-22 on page 206.

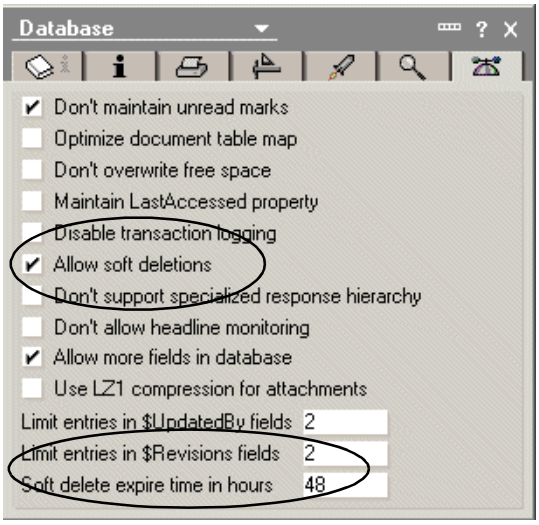



*Figure 8-22 Changing retention period of soft deletes*

### *Enabling quota enforcement*

1. From the Domino Administrator, click the **Configuration** tab, expand the Server section, and click **All Server Documents**.
2. Select the Server document to edit, and then click **Edit Server**.
3. Click the **Transactional Logging** tab. In the Quota enforcement field, select one of these methods. Then click **Save & Close**; see Figure 8-23.


*Figure 8-23 Quota enforcement field*

## 8.5 Upgrading an existing Domino server

This section describes upgrading a single Domino server using the script file method. Before you begin your upgrade, you should review the release notes and check the platform-specific section for any new features. Also review the fix list. There are also a couple of notes.ini parms which you may want to set before you begin your upgrade:

1. If you do not want to update your Domino Directory design (during an upgrade, for example), you deploy new designs in a centralized manner, then add the following notes.ini parm: `SERVER_UPGRADE_NO_DIRECTORY_REDESIGN=1`.
2. A Domino upgrade adds the “design” task in servertasks= line of the notes.ini. If you do not want to update your Server Tasks, add the following notes.ini parm:

   ```
   SetupLeaveServerTasks=1
   ```

### Upgrade steps

1. Shut down the server.
2. Back up the /opt/lotus files (library files and program files). If /opt/lotus is a mountpoint in your system, you might want to unmount the volume and mount a new one. See step 5.

3. Copy the following files from out of the notesdata directory into a temporary directory: cert.id, server.id, notes.ini, .profile, httpd.cnf, dcontroller.ini
4. Remove all files and directories from /opt/lotus.
5. Refer to the Install Guide to determine how much disk space is required in the program directory for the new release, and make sure that you have sufficient space allocated for this filesystem mounted on /opt/lotus.
6. FTP the tar file from the CD to /opt/lotus directory. Make sure that you use binary mode.
7. Telnet into the Linux as a root user and un-tar the file tar -xvzf <filename>. For example: **`tar -xvzf zLinux65xxx.TAR`**
8. Run the install program. We describe here how to upgrade the server using script-mode. If you want to do it interactively, see "Installing the executable server code" on page 138 for a detailed description.

   For either way, you will need the information gathered in Step 4 on page 198. In Appendix D, "Install script.dat" on page 385, there is an example of the script file for upgrading your Domino servers.
9. Make a copy of the script.dat, that is: **`copy script.dat domino65.dat`**. Change to the /opt/lotus/zlinux directory.
10. Update the following lines in domino65.dat. For example:

```
installation_type = 2 <Domino Enterprise Server>
program_directory = "/opt/lotus"
template_install_option = 0  <do not install template files>
data_UNIX_user = "domserva"
data_UNIX_group = "notes"
```

11. You will need one data_directories section for each Domino server:

```
data_directories: "/domserva/notesdata"
```

**Note:** Remove or add data_directories sections as required. For example, if your Linux has two Domino servers on it, you will have two data_directories sections.

12. # Uncomment either of the following lines to override the defaults:

```
data_UNIX_user  = "domserva"
data_UNIX_group = "notes"
```

13. Install the new code by entering the following command from the /opt/lotus/zlinux directory: install -script ./domino65.dat
14. Copy the files from step 2 back to notesdata.
15. Reinstall any third-party software that loads files into /opt/lotus, such as backup APIs, monitoring software, or virus scanning software. Refer to that product documentation for details.
16. Logon with the Domino user ID.
17. Bring up the server by running **`/opt/lotus/bin/server`** or **`/opt/lotus/bin/server -jc`** (to bring up Domino Controller).

**Important:** During an upgrade, you will be using your existing ID files and database, so you should *not* execute Domino Server setup.

18. Depending on your notes.ini setting (see step 1 on page 206), you might be prompted as to whether the design of the Domino Directory should be replaced or not.

## 8.6 Problem determination

If there are server problems or failures, you will want to restart the server as quickly as possible, but you will also want to collect the necessary documentation in order to resolve the problem. Setting fault recovery and running Notes System Diagnostic (NSD) will help you to accomplish these objectives.

### 8.6.1 NSD tool

Notes System Diagnostic (NSD) is a diagnostic script that gathers information. It is used to troubleshoot problems and verify that the server is configured correctly. NSD log files can be used as a tool to determine the cause of a server crash. NSD and memcheck are now bundled with the core Lotus Domino 6 product, and NSD is now the default debugger. The output of the NSD tool can be sent to Lotus Support to help diagnose server problems.

NSD output is in plain text and can be viewed with any text file viewer. It contains some basic configuration information and current processes running on the system (ps output). It also contains notes.ini and general system information (Linux version, local disks, and so on), as well as a memcheck portion.

#### NSD options

A number of options can be used with the NSD tool, depending on the level of detail required. Following are some of the options available:

| | |
|---|---|
| **-help** | Displays the nsd help list |
| **-info** | Report system info |
| **-memcheck** | Run the Notes memory checker only |
| **-kill** | Stops all Notes processes and cleanup-related IPC resources |

The command **`nsd -info`** will skip attaching to the processes with a debugger and obtaining a trace. This is useful when you are gathering only system information and do not need any process-level information for diagnosis.

Issue **`nsd -memcheck`** to run memcheck. Memcheck is a utility that obtains information about the current state of the Domino memory pools. It is installed by default in Domino 6 and is called by the NSD script, but it can also be run manually. For details about the memcheck command options, run **`memcheck -h`**

If Memcheck information is not needed, use the **`-nomemcheck`** option. This can reduce the total running time of the NSD script.

If you cannot shut down the server with a quit from the console, then **`nsd -kill`** should be run to ensure that the environment is clean for a server restart. Issuing **`nsd -kill`** will cancel all Notes processes and clean up IPC resources related to those processes.

The NSD report contains the following:

- Current processes
- Process tree
- netstat
- Notes.ini section
- System info section—Linux version, swap info, local disks, VM stats
- memcheck

## 8.6.2 Running NSD

When you encounter a hang condition on your server, it is advisable to run NSD so that you can send it to Lotus Support.You must be in the Domino data directory to run NSD, and you should run NSD as the Domino server Linux account.

The nsd file has information about the tasks which were running when the server crashed, as well as general system information. By default, nsd files are created in the IBM_TECHNICAL_SUPPORT directory located beneath the Notes/Domino data directory.

To change the directory where the nsd is created, set the following option in the notes.ini:

```
NSD_LOGDIR=/
```

To add which programs nsd should attach to and kill, set the following option in the notes.ini:

```
NSD_PATH=/
```

## 8.6.3 Fault recovery

We recommend that you enable fault recovery to automatically restart the server after a Domino server crash. The server will shut down, release all associated resources, and then restart automatically, without any administrator intervention. If you are using multiple partitions, only the partition which has the error is terminated and restarted. You can enable fault recovery in the server document under **Basics tab - Fault Recovery**, as shown in Figure 8-24.

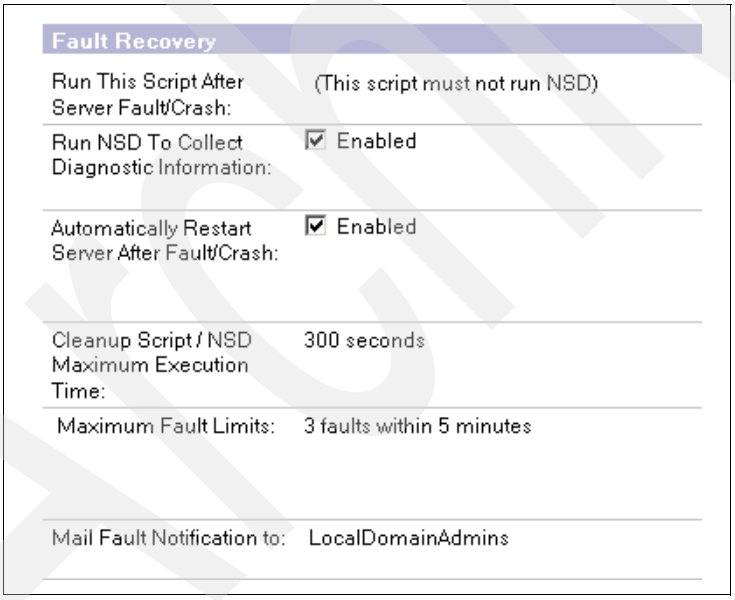



*Figure 8-24 Enabling fault recovery*

***Description of fault recovery settings***

| Field | Description |
|---|---|
| Run This Script After Server Fault/Crash | The name of an optional script that runs after a crash and before any other cleanup takes place. Enter the complete path and script name, including file extension. |
| Run NSD To Collect Diagnostic Information | Specifies whether to run NSD. |
| Automatically Restart Server After Server Fault/Crash | Specifies whether the server automatically restarts following a crash. |
| Cleanup Script/NSD Maximum Execution Time | Specifies the time, in seconds, that the cleanup script is allowed to run. If the script does not complete within the specified interval, it is stopped. The default execution time is 300 seconds (5 minutes). The maximum is 1800 seconds. |
| Maximum Fault Limits | The number of times the server is allowed to restart during a specified time period, in minutes (for example, 3 crashes within 5 minutes). If the number of crashes exceeds the number of allowed restarts for the interval, the server exits without restarting. |
| Mail Fault Notification to | The name of a user or group that Domino sends mail to after fault recover restarts the server. |

In summary, since NSD is invoked automatically and will collect the necessary PD information, it is not necessary to run an additional script. By enabling automatic restart, the server will automatically terminate all tasks and restart.

The default time of 300 seconds may not be enough time for NSD to complete on a larger server, so consider increasing this value. It is also a good practice to specify a group that gets notified when the server crashes.

## 8.6.4 CORE dump

Core dumps provide additional problem determination data to help resolve Domino problems. Enable core dumps (core.xxx) with the following notes.ini variable:

```
DEBUG_ENABLE_CORE=1
```

The default location for core dumps is in the Domino notesdata directory. Since they can be very large, it is recommended that core dumps be directed to another directory with more space. This can be done with the following notes.ini variable:

```
DEBUG_CORE_PATH=/path...
```

**Tip:** You can put core dumps and NSDs in the same directory to better manage these files.

## 8.6.5 Using Notes log

Every Domino server records information about server activities in the log database (log.nsf). This is helpful when you are doing problem determination on your system. Some of the information it includes:

- Database usage by user
- Database size
- Mail routing, replication and other events

- Usage of the system by user, including:
  - User name
  - Connect time
  - Number of documents read and written, by database
- Amount of data transferred across the network
- Number of transactions run

You can choose to collect replication and client session event records when you initially set up the server. You can change these settings later with the Log_Replication and Log_Sessions parameters in notes.ini.

### 8.6.6 Problem scenario

Let us assume there is a problem with a backlog of mail on the mail server, DomServA/ITSO. The Domino Administrator would issue **`tell router show`** from the console. We show the response from this command, divided into three sections, and include an explanation on how to interpret the output of each.

The first section of the output of **`tell router show`** is shown in Example 8-5.

*Example 8-5 tell router show command -partial output*

```
Msgs State     Via   Destination
    2 Retry(16) NRPC  [$LocalDelivery] mail/cb123lmt
                      Last error: File does not exist
                      Next retry: 08/28/2003 10:39:04
```

This example provides details on messages that are pending Local Delivery using NRPC. These messages are currently showing a RETRY message state, which means the messages cannot be delivered. Note the next line: `File does not exist`. This could indicate that the file is not physically there, or that there may be a problem with that file.

The [$LocalDelivery] Destination indicates these messages are for a local user on this server. The protocol being used to deliver these messages is NRPC. There are two messages in Retry state attempting to route with NRPC to [$LocalDelivery]. The number immediately following the State, 2 Retry(16) indicates the number of threads available to use for delivery of these messages. If these files are backed up and cannot make the local delivery, then the Domino Admin can confer with Linux Admin to check the iostat for the response time of the DASD and perhaps discover a bottleneck.

The second section of the output of the *tell router show* is shown in Example 8-6.

*Example 8-6 Continuation of tell router show command*

```
0 Retry(21) NRPC  CN=DOMSERVB/O=ITSO (Pull/Push)
                  Last error: The server is not responding. The server may be down
                  or you may be  experiencing network problems.
                  Contact your system administrator if this problem persists.
                  Next retry: 08/28/2003 11:04:40
    0 Wait      NRPC  CN=DOMSERVC/O=ITSO (Pull/Push)
    0 Retry(21) NRPC  CN=DOMSERVD/O=ITSO (Pull/Push)
                  Last error: The server is not responding. The server may be down or
                  you may be experiencing network problems.
                  Contact your system administrator if this problem persists.
                  Next retry: 08/28/2003 10:44:22
```

Immediately following the Message Status information, you will often see the reason why messages are in queue. If the message is in retry, along with Last Error message, you will also see the Next Retry Date and Time. The reason is because the destination server is not responding. This type of event would alert Domino Administrator to check with Linux Administrator to see if perhaps a network problem exists, or to see what is happening to the server that is not responding.

The third section of the output of **`tell router show`** is shown in Example 8-7.

*Example 8-7 Final extract - tell show router*

```
Transfer Threads: Max = 4; Total = 3; Inactive = 0; Max Concurrent = 2
Delivery Threads: Max = 4; Total = 1; Inactive = 0
```

The number of Transfer Threads is currently 4, 3 of which have already been spawned. One more will be started if additional threads are required. The Maximum Concurrent Transfer Threads is 2, which is half the number of configured Transfer Threads.

The router sets a default maximum number of transfer and delivery threads based on server memory. So here, if mail does appear slow, the statistics can aid in determining if the memory is sufficient to sustain the mail volume.

## 8.7 Domino Web Access (DWA)

Domino Web Access (DWA), formerly known as iNotes Web Access or just iNotes, provides Notes users with browser-based access to Notes mail and Notes calendaring and scheduling features. Using Domino Web Access, a user can send and receive mail, view the calendar, invite people to meetings, create to do lists, keep a notebook, and work offline.

To set up Domino Web Access, choose “Web Browsers (HTTP Web services)” during Server Setup. If you want to give users the ability to work offline, also choose Domino Off-line Services (DOLS). DOLS is not required to run Domino Web Access.

After being set up for Domino Web Access, a user can use both the standard Notes client and a Web browser to access their mail files. Because both the Notes client and Domino Web Access operate on the same underlying user mail file, read and unread marks remain up-to-date, regardless of which client is used to read the mail. Users can also synchronize contact information in their Personal Address Book with information in their Contact List in Domino Web Access.

While users simply need a name and Internet password to log on and use Domino Web Access, a Notes ID is required if a user wants to work offline and uses the Notes Client. If you want users to use their Notes Client or to work offline, be sure to create a Notes ID for each one when registering new users with the Domino Web Access template.

In the following list, we highlight some of the new features for DWA that come with Domino 6.5. For a complete list and a more comprehensive description, refer to the release notes of Domino 6.5.

- General
  - Mozilla support - beginning in 6.5, Domino Web Access supports Mozilla 1.3 on Red Hat Linux 7.2.
  - GZIP compression - beginning in 6.5, Domino Web Access uses compression (GZIP format) to reduce network bandwidth consumption and provide better performance, particularly for users with slow network connections.

- Customizing - Domino developers can customize the design of the iNotes template. Using Domino Designer, developers can modify the following forms to add action buttons to views or dialog boxes, provide additional choices for the Welcome Page, and replace the Domino Web Access logo with a corporate logo. For more information about this utility, see the release note “Domino Web Access Forms Customizing” in the section “Things You Need to Know - Domino Web Access”.

- Mail
  - Copy Into - when reading a mail message, users can copy the contents of the body of the message into a new calendar or to do entry. From the open mail message, click **Copy Into** and select **New Calendar Entry** or **New To Do**.
  - Mail database information - users can get information about the size and free space available on the mail database; in Domino Web Access Preferences - Other.
  - View unread messages - users now have a new view option that shows unread messages only.
  - Mark messages for follow-up - users can mark a message for follow-up. Priority is indicated with an icon, shown in the inbox view.
  - Secure mail - the ability to send, sign and verify Notes-encrypted messages.
  - Block sender - users can set new mail rule that blocks the mail from a specified sender.
  - Phone message - users can fill out a phone message form that is sent over e-mail to inform someone of a phone call.
  - Local archiving - users can choose to archive locally to their own workstations.
  - Internet formatting - users can choose to use Internet formatting when replying to a mail message.
- Calendar
  - Time stamp calendar printouts - added the time to calendar printouts so that users can tell which is the most recent between two versions printed on the same day.
  - Multiple time zones - the calendar supports the display of multiple time zones in the calendar.
  - Calendar and Scheduling delegation - users can delegate the scheduling and response to meeting invitations to another user.
- To Do
  - Users can create a group to do that can be assigned to one or more people or groups.
- Instant Messaging
  - Instant Messaging functionality has been incorporated into Domino Web Access. This means that users can link to other Sametime® users and open chat sessions without launching the Sametime application.

## 8.8 Best practices

1. Split Server by function - when you have more than one Domino server in your organization, we recommend that you dedicate each server to special task. That means you would have a server for Notes mailing, another for applications, one as hub or administration server, and a server for POP3.
2. Selective Clustering - a Domino cluster is a group of two or more servers that provides users with constant access to data, balances the workload between servers, improves

server performance, and maintains performance when you increase the size of your enterprise.

The servers in a cluster contain replicas of databases that you want to be readily available to users at all times. If a user tries to access a database on a cluster server that is not available, Domino opens a replica of that database on a different cluster server, if a replica is available. Domino continuously synchronizes databases so that whichever replica a user opens, the information is always the same.

You may selectively cluster in order to get high availability of important databases. When a hardware or software problem occurs, clustered servers redirect database open requests to other servers in the cluster to provide users with uninterrupted access to important databases. This process is called *fail-over*. Clusters provide fail-over for business-critical databases and servers. Fail-over also lets you perform server maintenance, such as hardware and software upgrades.

3. Regular cleanup
   a. It is recommended to compress core dumps using gzip to save diskspace - **`gzip [files]`**
   b. It is a good practice to regularly clean up old core dumps and NSDs, especially after they have been shipped to Lotus Support.
4. Disable transaction logging for the mail.boxes by issuing the command **`compact -t mail.box`**, or by disabling it in the database property box of the mailboxes, as shown in Figure 8-25.

**Note:** Remember that you have to do this every time a mail.box gets created!

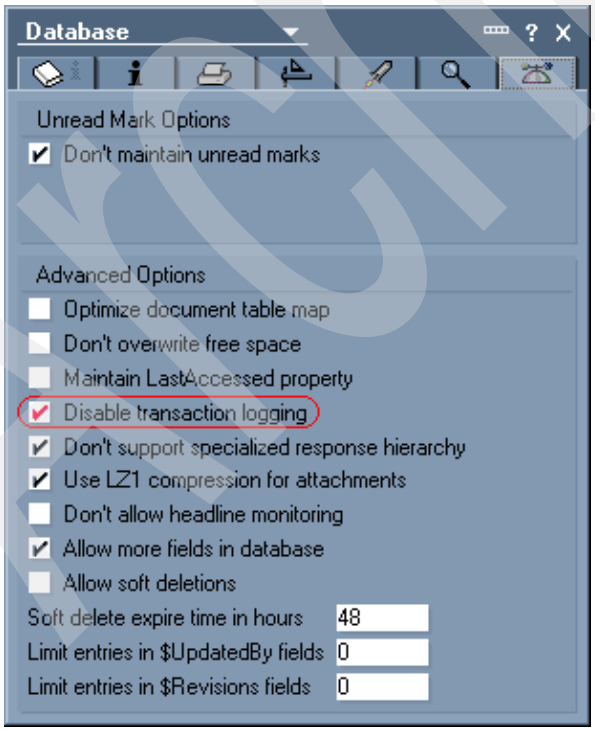



*Figure 8-25 Database properties for mail.box*

5. Implement at least two mail.boxes on your mail servers.

6. Review your notes.ini and remove any tasks that you will not be using.
7. Anti-spam for Mail environment. Refer to *Lotus Domino 6 Spam Survival Guide,* SG24-6930.
8. Define a service level agreement (SLA) with your customer (Notes users).

## 8.9 New 6.5 features

The following new features are listed in the Release Notes for 6.5 (readme.nsf file). In regard to the Domino Server specific to serviceability, the following enhancements include:

- Unify Fault Recovery/CleanupScriptPath interface - simply stated, these features can be set in the server document, as shown in 8.6.3, “Fault recovery” on page 209.
- Free-running memcheck to validate in-memory data structures (was initially run with nsd; see Example 8-8).

*Example 8-8 Free-running memcheck*

```
domserva@linuxa:/domserva/notesdata > memcheck

Copyright (c) IBM Corporation 1987, 2003. All Rights Reserved.

Usage: memcheck [options]
Options:
    [ -k c*urrent ]          -- analyze shared memory using default/current environment
    [ -k shmkey ]            -- analyze shared memory with IPC key 'shmkey'
    [ -p pid ]               -- analyze process 'pid' heap memory
    [ -c corefile ]          -- analyze core file 'corefile' memory
    [ -f imagedump ]         -- analyze memory saved in file 'imagedump'

  Examples:
    memcheck -k 0xf8000000
    memcheck -p 1251 -p 1255
    memcheck -c core
    memcheck -k 0xf8000000 -p 1254 -d all -o imagedump.out
    memcheck -f imagedump.out -v2
    memcheck -k 0xf8000000 -t 0x4120   # show block type 0x4120 entries only
    memcheck -k 0xf8000000 -t 0x4100   # show package 0x4100 entries only
```

- Timestamps in SEMDEBUG.TXT (Semaphore timeout issues): If a semaphore timeout occurs, this can be reported as SemTimeouts at the console (sh stat sem.timeouts), or you may see the following message: `session semaphore held for [n] seconds`. Technical Support will request that you capture the text file containing the semaphores. Previously there was no way of telling the date or time; that has been changed in this release.

The new features listed in Table 8-8 on page 216 are derived from the *Administration Help* database (help65_admin.nsf).

*Table 8-8 New Domino 6.5 features*

| New Feature | Description | For more information |
|---|---|---|
| Server setup and installation<br>New scriptable client setup variables | When setting up Notes with a scriptable setup, you can now include Sametime configuration variables in that setup. Notes users can then configure Sametime for use within their Notes client. | See the topic *Setting up Notes with a scriptable setup*. |
| Server.Load<br>New SendMessage profile command for Server.Load | The SendMessage profile command creates and sends messages of varying sizes according to a pre-established set of sizes and distribution percentages | See the topic *SendMessage command and SendMessage profile command* |
| Server Monitoring<br>Support for platform statistics for Domino on Linux for zSeries and Linux | Platform statistics are now supported for Linux and for Linux on zSeries. You can now view, control the reporting of Linux platform statistics, collect statistics, and perform other statistics-related tasks just as you would for other Domino server platforms. | See *Monitoring the Domino system* for more information. For detailed information regarding statistics on the different platforms, see *Evaluating platform statistics* |
| Administration<br>Delete in Unread Lists administration request added to the Delete Person in Domino Directory sequence of requests. | When the Delete in Unread Lists administration request runs, each server in the domain examines every database on the server and updates the person's name in any unread lists. | See *Delete person in Domino Directory* |
| Resource Reservations<br>Rename resources | You can rename a Resource by changing its name, Site, and, if the Resource is of type “Other,” its category. | See *Editing and deleting Resource documents* |
| SNMP Integration<br>PEER SNMP agents are replaced<br>with the NET-SNMP Master Agent | The PEER Networks SNMP Master Agent has been replaced with the NET-SNMP Master Agent on Solaris. The NET-SNMP Master Agent is also now provided on Linux. | See *Using NET-SNMP with the Domino SNMP Agent.* |

## 8.9.1 Policies

Policies have not changed much from 6.x to 6.5 in Domino. We offer a brief description of using a policy and how you control the way users work with Notes. A *policy* is a document that identifies a collection of individual policy settings documents. Each of these policy settings documents defines a set of defaults that apply to the users and groups to which the policy is assigned. Once a policy is in place, you can easily change a setting and it will automatically apply to those users to whom the policy is assigned.

Policy settings documents cover these administrative areas:

**Registration** If a policy including registration policy settings is in place before you register Notes users, these settings set default user registration values

including user password, Internet address format, roaming user designation, and mail.

**Setup** If a policy (including setup policy settings) is in place before you set up a new Notes client, these settings are used during the initial Notes client setup to populate the user's Location document. Setup settings include Internet browser and proxy settings, applet security settings, and desktop and user preferences.

**Desktop** Use desktop policy settings control and update the user's desktop environment or to reinforce setup policy settings. For example, if a change is made to any of the policy settings, the next time users authenticate with their home server, the desktop policy settings restore the default settings or distribute new settings specified in the desktop policy settings document.

**Mail archiving** Use archive policy settings to control mail archiving. Archive settings control where archiving is performed and specify archive criteria.

**Security** Use security settings to set up administration ECLs and define password-management options, including the synchronization of Internet and Notes passwords.

**Important to note:**

- Delete Policies and Subpolicies from previous releases. If you created Master Policies or Subpolicies (now “Settings”) in an earlier release of Domino 6, you must delete them from the Domino Directory before updating the Domino Directory design with the current Domino 6 template.
- Subpolicies have been renamed to “settings”, and field names have been changed. Master policies are now called “policies”.
- Setup policies are not supported for roaming users.
- User preferences will not be set correctly when roaming from one machine to another if the user uses policy settings for their initial setup.

The following figures illustrate the policy that we created for our domain to enable network compression. First we created a policy, DomServA Policy, as follows:

1. From the Domino Administrator, click the **Configuration** tab, then open the Policies view.
2. Click **Create Policy**, which will display the policy screen, as shown in Figure 8-26 on page 218.

If this is your first policy, you will need to define policy settings (see Figure 8-27 on page 218).

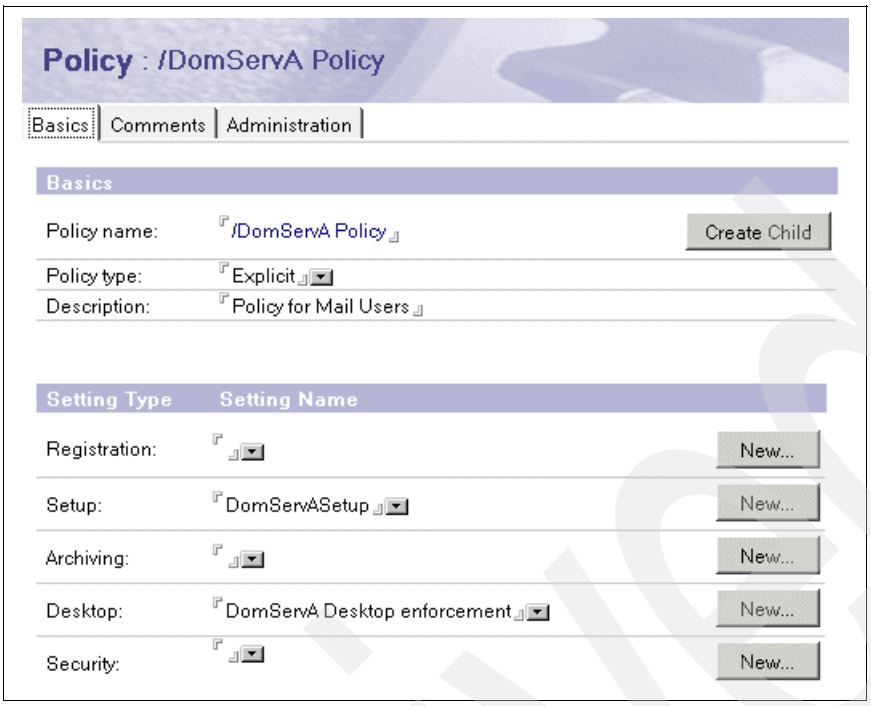



*Figure 8-26 Create Policy*

Under this policy, we then created the setup policy setting and the desktop setting. This policy setting is applicable to all mail users, and we wanted to enforce compression over the network.

Figure 8-27 shows one example that administrators can use to make sure all mail users in the environment have the same settings. Also by using the setup policy, we can make sure new registered users will be created with the desktop we enforced for existing users.

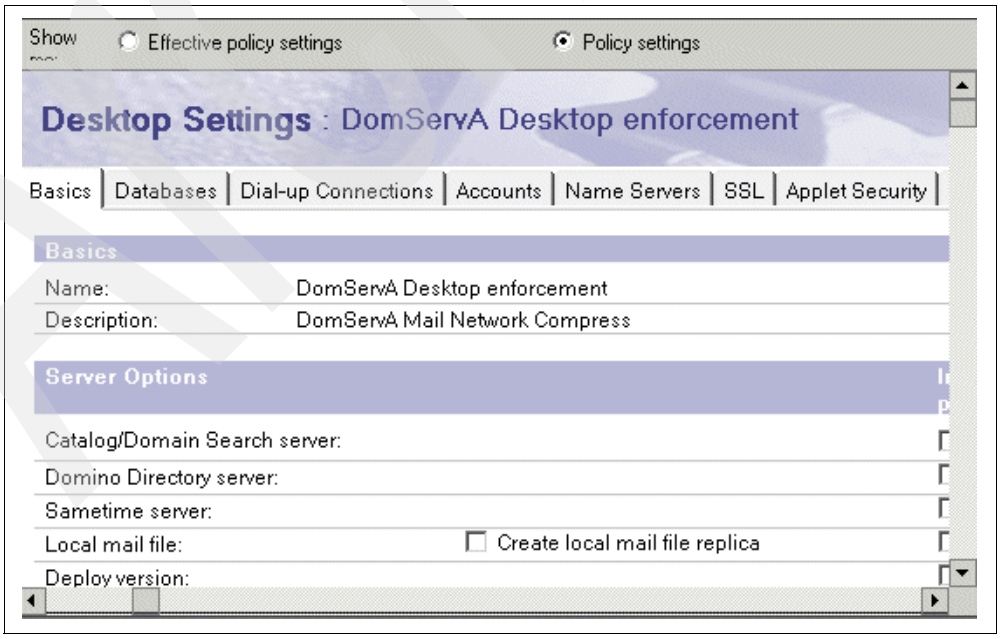



*Figure 8-27 Desktop policy setting*

Figure 8-28 shows the policy by hierarchy view after the policy has been created.

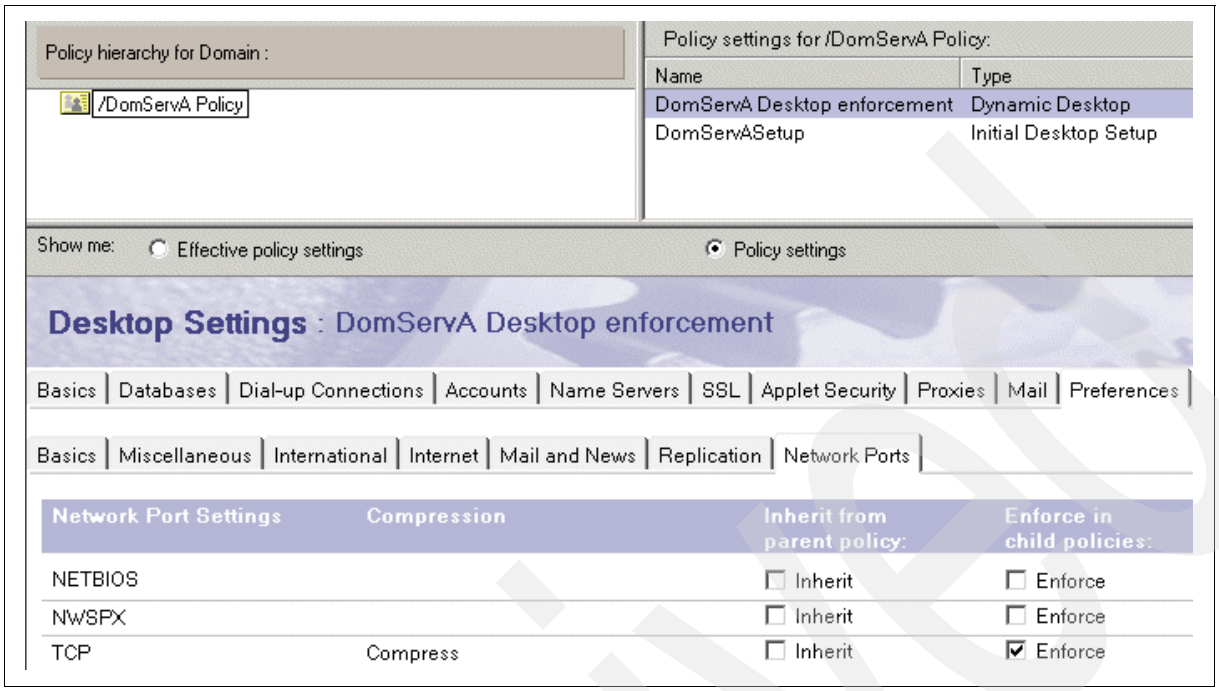



*Figure 8-28 Policy by hierarchy view*

## 8.9.2 Directory synchronization

The Active Directory Synchronization tool, or ADSync, allows Active Directory administrators to manage (register, delete, and rename) users and groups in both Active Directory and the Domino Directory as a unified operation from the Active Directory Users and Computers Console. A Redpaper is available on this tool, entitled *Active Directory Synchronization with Lotus ADSync*, REDP-0605-00

9

# Systems management

In this chapter, we discuss administration or systems management topics from the Linux and VM system programmer points of view. The topics include Domino user administration, monitoring, performance monitoring, and backup and recovery. This chapter does not present a comprehensive discussion for Linux or z/VM systems themselves, but we touch on those topics of special interest to someone managing Domino servers.

For a discussion of topics from the Domino administrator point of view, see Chapter 8, “Domino administration” on page 173.

For backup and recovery, see Appendix B of the Tivoli Storage Manager for Mail 5.2 publication *Data Protection for Lotus Domino for UNIX and OS/400 Installation and User's Guide,* SG32-9056.

# 9.1 Linux administration

Administering Linux on a zSeries system includes managing the filesystems (DASD, minidisk, XPRAM, and swap space), managing users, automating various tasks, and starting and stopping the Linux servers. We do not cover all these topics in this redbook, but in this section we discuss the aspects of Linux administration that are related to managing a Domino system. In Chapter 9 of *Linux for S/390*, SG24-4987, you'll find an excellent general discussion of managing the filesystem and users.

# 9.2 Managing the filesystem

In Chapter 4, "Disk configuration" on page 53 there is a discussion of the different filesystems available with Linux. In this section we address the types of commands Linux administrators might need to use in their day-to-day support of the system.

## 9.2.1 Formatting a device

Before the physical volumes can be created, they must first be formatted and partitioned for Linux, whether running under VM or native. This is done using the **dasdfmt** and **fdasd** commands.

- To format a single DASD with the recommended block size of 4 KB, use:

```
dasdfmt -b 4096 -f /dev/dasdg
```

- The **fdasd** command allows you to split a DASD into several partitions. To create a single partition on formatted DASD, use the **fdasd** command with the **-a** option:

```
fdasd -a /dev/dasdg
Note: -a option is to auto-create one partition in non-interactive mode
```

If running Linux under VM, you can use the CMS format and the **RESERVE** command to prepare the physical device under VM. The VM minidisk device driver can only be used with CMS-format minidisks.

- To format a minidisk, use the CMS command:

```
FORMAT <device number> <filemode> (blksize <blksize>
for example: FORMAT 202 D
```

- Use the **RESERVE** command to allocate all available blocks of a formatted CMS minidisk to a unique CMS file:

```
RESERVE linux mdisk <filemode>
for example: RESERVE LINUX DIAG D
```

This is what is recommended for minidisks, since the CMS label/volid information is preserved, provided the filesystem is created on a Linux device partition.

> **Note:** Use of the **dasdfmt** command is *not* recommended for minidisks, since it will destroy any CMS label/volid information.

## 9.2.2 Creating a filesystem

The **mke2fs** command is used to create an ext2/ext3 filesystem (usually in a disk partition). To create an empty ext2 filesystem after formatting the device, use the command **mke2fs**.

Here is the syntax:

```
mke2fs [-b <block size>] /dev/dasd<n>1 | /dev/xpram<n>
for example: mke2fs -b 4096 /dev/dasdg
```

Use the **-j** option to create the filesystem with an ext3 journal. Note that you must be using a kernel which has ext3 support in order to actually make use of the journal. Our SLES-8 system had the appropriate support for ext3 journaling. For example, to create an ext3 filesystem on lv01, issue:

```
mke2fs -j -b 4096 /dev/vg01/lv01
```

### 9.2.3 Mounting a filesystem

A *mount point* is a directory that is usually empty. If it is not empty, any files it might contain are hidden as long as a filesystem is mounted over it. Manual mounting is done with the mount command:

```
mount -t type dev mountpoint
```

The following command mounts the filesystem that is on /dev/dasdg1 over the directory /mnt:

```
mount -t ext2 /dev/dasdg1 /mnt
```

**Note:** To automatically mount a filesystem on startup, add an entry in /etc/fstab.

### 9.2.4 Managing Logical Volume Manager (LVM)

For an extensive overview of LVM, refer to Chapter 4, “Disk configuration” on page 53. The following are some common commands that you can use when managing your LVMs.

- To initialize a physical volume, use **pvcreate**:

  ```
  Syntax: pvcreate /dev/hdb
  For example: pvcreate /dev/dasdl /dev/dasdm1
  ```

- To create a volume group, use **vgcreate**:

  ```
  Syntax: vgcreate my_volume_group /dev/hda1 /dev/hdb1
  ```

  For example, to create the volume group named vg01, you issue:

  ```
  vgcreate vg01 /dev/dasdl1 /dev/dasdm1
  ```

- To display the attributes of volume groups such as size, use **vgdisplay**:

  ```
  Syntax: vgdisplay my_volume_group
  For example: vgdisplay /dev/vg01
  ```

- To add physical volumes to a volume group, use **vgextend**. To add an initialized physical volume to an existing volume group:

  ```
  vgextend my_volume_group /dev/hdc1 where hdc1 is the new physical volume
  ```

- To extend a logical volume, use **e2fsadm**. LVM has a utility called e2fsadm to allow resizing of a logical volume containing an ext2 or ext3 filesystem:

  ```
  Syntax: e2fsadm -L xxG /dev/myvg/homevol
  ```

  For example, to extend /dev/vg01/lv01 filesystem using VG, issue:

  ```
  e2fsadm -L 20G /dev/vg01/lv01
  ```

**Important:** You will still need to unmount the filesystem before running e2fsadm. This means the Domino server must be stopped.

## 9.2.5 Monitoring the filesystem size

You can use the **df** command to monitor disk usage and to determine if you need to increase the capacity of your Domino server filesystems. You will need to work together with the Domino Administrator if you need to expand a filesystem (such as extending an LVM), because the Domino Server has to be shut down and the filesystem unmounted to perform this function.

You will also want to check on a regular basis to see if your filesystems are getting full. To monitor your filesystems for a specific threshold (for example, 75%), there are Tivoli and other third-party tools that you can use. Refer to *Linux on IBM server zSeries and S/390: System Management,* SG24-6820 for more information about this topic. This can also be done with Domino tools; for more information refer to 8.3.1, “System monitoring tools” on page 186.

### df

The **df** command displays information about the filesystem, including the total amount in the filesystem, the amount used, and the amount available. Example 9-1 shows the output from the **df** command.

*Example 9-1 df command*

```
root@linuxa:/ > df
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/dasdb1            2259188   1067352   1191836  48% /
/dev/dasdc1            2403184   1538872    864312  65% /opt
shmfs                   257140         0    257140   0% /dev/shm
/dev/dasdt1            2365444   1453272    792012  65% /domserva
/dev/mail1/mail1      16509864    216116  15455092   2% /domserva/notesdata/mail1
/dev/mail2/mail2      16509864        20  15671188   1% /domserva/notesdata/mail2
/dev/translog/translog 4313840   2099480   1995224  52% /domserva/notesdata/translog
```

Example 9-2 shows **df** with the **-h** option from our test server. **df -h** shows the space in more “human readable” format (size is displayed in M or G).

*Example 9-2 df -h command*

```
root@linuxa:/ > df -h
Filesystem           Size  Used Avail Use% Mounted on
/dev/dasdb1          2.2G  1.1G  1.2G  48% /
/dev/dasdc1          2.3G  1.5G  845M  65% /opt
shmfs                252M     0  252M   0% /dev/shm
/dev/dasdt1          2.3G  1.4G  774M  65% /domserva
/dev/mail1/mail1      16G  212M   15G   2% /domserva/notesdata/mail1
/dev/mail2/mail2      16G   20K   15G   1% /domserva/notesdata/mail2
/dev/translog/translog4.2G  2.1G  2.0G  52% /domserva/notesdata/translog
```

You can also display the filesystem usage (as shown in Figure 9-1 on page 225) with the Resource Measurement Facility Performance Monitoring (RMF™ PM) tool, which we describe in 9.3.2, “Performance monitoring with RMF PM for Linux” on page 232.

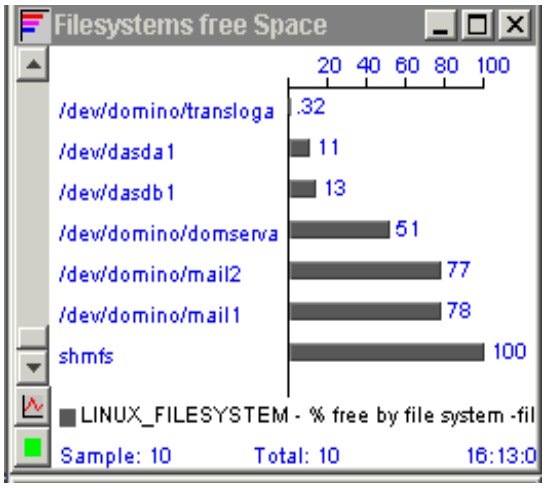



*Figure 9-1 Filesystem free space display using RMF-PMS*

## 9.2.6 Managing users

For user ID administration from the command line, you can run **useradd** or **userdel** to add or delete a user. Or you can use a graphical desktop environment, such as SuSE's YaST2 or KDE User Manager, to manage your accounts. Only the Domino servers need to have Linux user accounts; end users will be defined in the Domino Directory using the Domino Administrator client.

### Adding users

You can create a user account using the command **useradd**. This command creates a new user. It is in the /usr/sbin directory, which you might need to add to your PATH:

```
Syntax: useradd [-d <home>][-g <group>] [-G additional groups>] [-m]
[-s <shell>] [-u <uid>] [-p passwd] <username>
```

For example, to add a user named user1, issue the command:

```
useradd -m -d /home/user1 -p dummypw -c 'John Smith' user1
```

This user will have the password dummypw, which can be changed using the passwd command. An entry will be created in the file /etc/passwd.

The following is the format of the passwd entries file:

user:password:uid:gid:description:home:login_shell

### Deleting users

The **userdel** command deletes a user; here is the syntax:

```
userdel -r user2
```

### Changing ownership and permissions

It is important that the Domino data files, particularly the notesdata directory be owned by the Linux account of the Domino server. If you need to change the permissions (mode) of files or directories, you can use the **chmod** command.

```
chmod [-r] permissions filename
i.e. chmod 755 filename
```

You may also need to change the owner or group that is assigned to a file or directory. This can be done using the **chown** command.

```
chown newowner:group directory_path
```

For example, if you create a new filesystem and mount it, you may have to change the permissions to make the Domino server user ID as the owner of that directory.

```
chown domserva:notes domserva/notesdata
```

You can also change in the subdirectories of the directory that you are currently in "recursively" by using the **-R** option:

chown -R *userid directory_path*

## 9.2.7 Scheduling of jobs with crontab

Shell scripts can be used to manage your Linux server. The crontab scheduler in Linux automates the process of running these scripts. You can have several scripts for maintaining your server, and you can use crontab to run them at specified times. To list the scheduled programs in crontab, log in as root and issue **crontab -l**.

For example, it is a good practice to clean up your core dumps on a regular basis so that your filesystems do not fill up. You can write a script to do this and schedule it to run every day.To run a cleanup script every day at 1 a.m., set up the following cron job:

```
0 1 * * * ~/cleanup.sh 1>/dev/null 2>&1
```

**Note:** The ~ symbol refers to the home directory.

## 9.2.8 System logs

Linux keeps logs in /var/log unless the administrator changes the path. The daemon (program) responsible for generating the logs is syslogd. Log entries are caused by events, and almost every application can send information (events) to the syslogd. The default is to start the syslogd daemon when Linux starts. Also by default, all system messages go in the /var/log/messages file. A sample messages file is shown in Example 9-3.

The log files can be redirected to other paths by editing the syslog.conf or by moving the file and creating a link to the new location. In order to see the log information in real time, log in as root and at the shell command prompt, issue **tail -f /var/log/messages**, as we did in Example 9-3. This **tail** command will display the last few lines the log file and continue to display new lines until you exit (usually with Ctlr+C).

*Example 9-3 Messages log*

```
root@linuxa:/ > tail -f /var/log/messages
Aug 12 09:56:59 linuxa sshd[16338]: Failed password for root from ::ffff:1.23.4.5 port 1179
Aug 12 09:57:03 linuxa sshd[16338]: Accepted password for root from ::ffff:1.23.4.5 port 1179
Aug 12 09:59:00 linuxa /USR/SBIN/CRON[16417]: (root) CMD ( rm -f /var/spool/cron/lastrun/cron.hourly)
Aug 12 10:00:00 linuxa /USR/SBIN/CRON[16442]: (root) CMD ( /usr/lib/sa/sa1      )
Aug 12 10:10:00 linuxa /USR/SBIN/CRON[17347]: (root) CMD ( /usr/lib/sa/sa1      )
Aug 12 10:20:00 linuxa /USR/SBIN/CRON[17395]: (root) CMD ( /usr/lib/sa/sa1      )
Aug 12 10:30:00 linuxa /USR/SBIN/CRON[17426]: (root) CMD ( /usr/lib/sa/sa1      )
Aug 12 10:33:20 linuxa sshd[17457]: Accepted password for domserva from ::ffff:1.23.4.5 port 1184
Aug 12 10:33:20 linuxa sshd[17459]: fatal: PAM session setup failed[6]: Permission denied
Aug 12 10:33:39 linuxa sshd[17462]: Accepted password for root from ::ffff:1.23.4.5 port 1185
```

### 9.2.9 Remote administration

Linux servers can be administered remotely, and many tools are available to do this. Webmin and VNC are two tools that are described in detail in *Domino 6 for Linux*, SG24-6835.

## 9.3 Monitoring

In this section we briefly describe some monitoring tools which Linux or z/VM systems programmers might use to manage the Linux servers where Domino is running. For more information, we have included references to several other IBM Redbooks. In addition, you can find more information about monitoring the Domino servers with Domino tools and functions in Chapter 8, “Domino administration” on page 173.

### 9.3.1 Linux tools

There are standard tools that come with the Linux system, as well as tools available from IBM and third-party vendors, that can be used to monitor your system and check the status of the Domino server. In this section, we describe some of tools (top, vmstat, sar, ps, and ipcs) that come with the United Linux 1.0 system distribution.

#### Top

The **top** command gives you a comprehensive look at what your Linux system is doing. It shows a table of active processes that is continually updated. The processes using the most CPU are displayed at the head of the list.

Top lets you interactively send signals to, or kill, individual processes. Its behavior, including update intervals and sorting order, can be adjusted, either interactively or by using startup options.

To run top, enter **top**. The resulting screen (Example 9-4) shows you the current time, how long the Linux server has been up, how many users are logged in, and how many processes are running. You also see how memory and swap space are being used.

*Example 9-4 The top command*

```
10:19am  up 18:41,  9 users,  load average: 1.43, 1.83, 2.36
246 processes: 242 sleeping, 4 running, 0 zombie, 0 stopped
CPU0 states: 15.0% user,  6.1% system,  0.0% nice, 78.5% idle
CPU1 states: 11.3% user, 28.2% system,  0.0% nice, 60.0% idle
Mem:   255220K av,  252240K used,    2980K free,       0K shrd,    1896K buff
Swap:  247760K av,   90428K used,  157332K free                  198460K cached

  PID USER     PRI  NI  SIZE  RSS SHARE STAT %CPU %MEM   TIME COMMAND
29036 domserva  15   0 99660  94M 95720 S    21.9 37.8   6:38 compact
   17 domserva  21   0  1168 1168   840 R    21.0  0.4   0:22 top
32633 domserva  15   0  8972 6832  1240 S     3.6  2.6   0:23 java
26557 domserva  15   0  8972 6832  1240 S     3.4  2.6   2:03 java
    8 root      15   0     0    0     0 SW    1.7  0.0   3:00 kswapd
    6 root      34  19     0    0     0 SWN   1.1  0.0   1:12 ksoftirqd_CPU0
26536 domserva  15   0  8972 6832  1240 S     0.7  2.6   0:32 java
27163 domserva  15   0 41964  34M 33012 S     0.4 13.6   0:00 server
    7 root      34  19     0    0     0 SWN   0.3  0.0   3:33 ksoftirqd_CPU1
27222 domserva  15   0 67596  63M 64004 S     0.3 25.2   0:04 sched
27113 domserva  15   0 41964  34M 33012 S     0.1 13.6   0:00 server
28611 domserva  15   0 48472  44M 45260 S     0.1 17.8   0:00 cldbdir
```

```
  1 root      15   0    76   60    36 S     0.0  0.0   0:01 init
  2 root      OK   0     0    0     0 SW    0.0  0.0   0:00 migration_CPU0
  3 root      OK   0     0    0     0 SW    0.0  0.0   0:00 migration_CPU1
  4 root      25   0     0    0     0 SW    0.0  0.0   0:00 kmcheck
  5 root      15   0     0    0     0 SW    0.0  0.0   0:00 keventd
  9 root      15   0     0    0     0 SW    0.0  0.0   0:24 bdflush
```

To exit top, press Enter q.

To see processes by CPU utilization, type **p**. To view information about a single user like the Domino server, type **u** and you will be prompted for the user ID; or you can press Enter for all users. Example 9-5 shows an example of the processes running under our Domino server ID domserva.

*Example 9-5 TOP display*

```
PID USER     PRI  NI  SIZE  RSS SHARE STAT %CPU %MEM   TIME COMMAND
  632 domserva  20   0   332  264   164 S     0.0  0.0   0:00 su
  633 domserva  15   0   444  208    36 S     0.0  0.0   0:00 bash
 7410 domserva  21   0  1476 1384  1268 S     0.0  0.2   0:00 bash
 7483 domserva  15   0 17888  17M  6688 S     0.0  3.4   0:03 java
 7495 domserva  15   0 17888  17M  6688 S     0.0  3.4   0:00 java
 7496 domserva  15   0 17888  17M  6688 S     0.0  3.4   0:00 java
 7497 domserva  15   0 17888  17M  6688 S     0.0  3.4   0:00 java
 7498 domserva  15   0 17888  17M  6688 S     0.0  3.4   0:00 java
 7499 domserva  15   0 17888  17M  6688 S     0.0  3.4   0:00 java
 7500 domserva  15   0 17888  17M  6688 S     0.0  3.4   0:00 java
 7503 domserva  25   0 17888  17M  6688 S     0.0  3.4   0:03 java
 7504 domserva  15   0 17888  17M  6688 S     0.0  3.4   0:00 java
 7505 domserva  15   0 17888  17M  6688 S     0.0  3.4   0:00 java
 7506 domserva  15   0 17888  17M  6688 S     0.0  3.4   0:00 java
 7507 domserva  15   0 17888  17M  6688 S     0.0  3.4   0:00 java
 7508 domserva  15   0 17888  17M  6688 S     0.0  3.4   0:00 java
 7509 domserva  15   0 17888  17M  6688 S     0.0  3.4   0:00 java
```

On a lightly-loaded system such as our test server, **top** often appeared at the top of the list when sorted by CPU utilization. You can set the frequency with which **top** updates the screen by issuing **s**. Even then, since **top** has a fairly large footprint in terms of memory and CPU, we recommend you do not allow it to run continuously. Instead, use it when you think the system is running slowly, to determine which processes are using the most CPU.

To learn more about **top** options, press **h** while **top** is running.

If you prefer a graphical view, there are alternatives to **top** in the KDE or GNOME desktops. In KDE, select **Process Management** from the Utilities menu. The GNOME desktop tool is System Monitor.

## vmstat

The **vmstat** command reports virtual memory statistics. It is one of the simplest, but most useful tools, because it reports important data about CPU, memory utilization, and disk I/O. The format of the **vmstat** command is:

```
vmstat interval_seconds number_of_intervals
```

The **vmstat** command without options displays the averages since system reboot. For example:

```
domserva@linuxa:~> vmstat
   procs                      memory    swap          io     system         cpu
```

```
 r  b  w   swpd   free   buff  cache  si  so    bi    bo   in    cs  us  sy  id
 0  0  0   1760   4824  34320 324736   0   0    90    91    0  1241   4   4  92
```

**vmstat** with options is shown in Example 9-6. We get a sampling of memory use every 10 seconds.

*Example 9-6 vmstat output*

```
domserva@linuxa:~> vmstat 10 5
   procs                      memory    swap          io     system         cpu
 r  b  w   swpd   free   buff  cache  si  so    bi    bo   in    cs  us  sy  id
 0  0  0  88464   3036   3404 190720  24   8   229   148    0   246  38  11  51
 0  0  0  88464   3024   3412 190720   0   0     0    12    0  1745   0   0 100
 0  0  0  88464   3012   3420 190720   0   0     0     2    0  1712   0   0  99
 1  0  0  88440   3048   3436 190744   2   0     2     8    0  1685   1   1  98
 0  0  0  88432   3032   3444 190744   0   0     0     2    0  1738   0   0 100
```

## sar

The **sar** (system activity reporter) command collects and reports information about system activity, including CPU, paging, file I/O, network, and so on. This command displays records previously saved in a file specified by a flag. By default, it displays the standard system activity daily data file, /var/log/sa/sadd file, where the dd parameter indicates the current day.

You can select information about specific system activities using flags. Not specifying any flags selects only CPU activity. You can save the output of the **sar** command into a file for later processing by another application. Records are time-stamped. Depending on the flags used, the output file can be saved in a format that can be easily read by a relational database system or by a pattern processing command like **awk**. Example 9-7 shows output from the **sar** command.

*Example 9-7 sar output*

```
domserva@linuxa:~> sar
sar
Linux 2.4.19-4suse-SMP (linuxa)         08/21/2003
05:50:00 AM       CPU     %user     %nice   %system     %idle
06:00:01 AM       all     49.70      0.00      9.74     40.56
06:10:00 AM       all     50.17      0.00      9.43     40.40
06:20:00 AM       all     49.74      0.00      9.79     40.48
06:30:00 AM       all     49.72      0.00      9.84     40.44
06:40:00 AM       all     49.70      0.00      9.75     40.55
06:50:00 AM       all     49.06      0.00     10.87     40.06
07:00:01 AM       all     49.46      0.00     10.03     40.51
07:10:00 AM       all     48.45      0.00     11.20     40.35
07:20:00 AM       all     49.83      0.00      9.87     40.30
07:30:00 AM       all     48.49      0.00     11.29     40.22
07:40:00 AM       all     49.12      0.00      9.70     41.19
```

## ps and pstree

The **ps** command displays a list of the currently running processes. Here are several examples.

**ps -ef** Displays a list of all currently running processes with full details, including user, process ID, start time, time, and program.

Using **ps -ef**, you can check which processes are running in the Domino server, as shown in Figure 9-2 on page 230.

```
domserva@linuxa:~> ps -ef |grep domserva
domserva  2419  2225  0 Aug21 pts/0    00:00:00 su domserva
domserva  2420  2419  0 Aug21 pts/0    00:00:00 bash
domserva 18513  2420  0 07:51 pts/0    00:00:00 bash
domserva 18586 18513  0 07:51 pts/0    00:00:02 /opt/lotus/notes/latest/zlinux/jvm/bin/exe/java -Xoss5M
-Djava.compiler=NONE -DDISPLAY= -cp /opt/lotus/notes/latest/zlinux/dconsole.jar
lotus.domino.console.DominoController -jc -c
domserva 18626 18613  0 07:51 pts/0    00:00:00 /opt/lotus/notes/latest/zlinux/logasio NOTESLOGGER reserved
domserva 18635 18618  0 07:51 pts/0    00:00:00 /opt/lotus/notes/latest/zlinux/server -j
domserva 18636 18618  0 07:51 pts/0    00:00:01 /opt/lotus/notes/latest/zlinux/server -j
domserva 18675 18613  0 07:51 pts/0    00:00:00 /opt/lotus/notes/latest/zlinux/event
domserva 18697 18675  0 07:51 pts/0    00:00:00 /opt/lotus/notes/latest/zlinux/event
domserva 18705 18697  0 07:51 pts/0    00:00:00 /opt/lotus/notes/latest/zlinux/event
domserva 18706 18618  0 07:51 pts/0    00:00:00 /opt/lotus/notes/latest/zlinux/server -j
domserva 18707 18618  0 07:51 pts/0    00:00:00 /opt/lotus/notes/latest/zlinux/server -j
domserva 18708 18618  0 07:51 pts/0    00:00:00 /opt/lotus/notes/latest/zlinux/server -j
domserva 18709 18618  0 07:51 pts/0    00:00:00 /opt/lotus/notes/latest/zlinux/server -j
domserva 18710 18618  0 07:51 pts/0    00:00:00 /opt/lotus/notes/latest/zlinux/server -j
domserva 18711 18618  0 07:51 pts/0    00:00:00 /opt/lotus/notes/latest/zlinux/server -j
domserva 18857 18710  0 07:51 pts/0    00:00:00 /opt/lotus/notes/latest/zlinux/update
domserva 18859 18710  0 07:51 pts/0    00:00:05 /opt/lotus/notes/latest/zlinux/replica
domserva 18864 18710  0 07:51 pts/0    00:00:00 /opt/lotus/notes/latest/zlinux/router
domserva 18866 18710  0 07:51 pts/0    00:00:00 /opt/lotus/notes/latest/zlinux/amgr
domserva 18881 18710  0 07:51 pts/0    00:00:03 /opt/lotus/notes/latest/zlinux/adminp
domserva 18883 18710  0 07:51 pts/0    00:00:00 /opt/lotus/notes/latest/zlinux/calconn
domserva 18886 18710  0 07:51 pts/0    00:00:01 /opt/lotus/notes/latest/zlinux/sched
domserva 19045 18866  0 07:51 pts/0    00:00:00 /opt/lotus/notes/latest/zlinux/amgr -e 1
domserva 19161 19055  0 07:51 pts/0    00:00:00 /opt/lotus/notes/latest/zlinux/cldbdir
domserva 19878 19018  0 08:21 pts/0    00:00:00 /opt/lotus/notes/latest/zlinux/replica
```

*Figure 9-2 Output of the ps -ef command*

**ps -U userid** Displays the currently running processes associated with that user. For example, **`ps -f -U domserva`** displays a detailed list of all processes related to our Domino server domserva.

**ps -efww** The ww option displays the command line options of each process.

**ps --help** Displays help for the **`ps`** command.

For more information about the ps command, enter man page on your Linux system.

The **`pstree`** command displays these same processes, but in a tree format starting with the first process called init. Here are several examples.

**pstree** Displays all processes starting at init.

**pstree -l** Wraps the lines instead of truncating them. This is handy if your display window is truncating the right side of the tree.

**pstree userid** Displays the portion of the tree related to that user. To see the processes for our Domino server in Figure 9-3 on page 231, we entered **`pstree domserva`**.

```
domserva@linuxa:~> pstree domserva
bash

bash---bash---java---java-+-24*[java]
                          `-java---server-+-event---event---8*[event]
                                          |-logasio---logasio---2*[logasio]
                                          `-server-+-70*[server]
                                                   |-server-+-adminp---adminp--+
                                                   |        |-amgr-+-amgr---3*[+
                                                   |        |      `-amgr---amg+
                                                   |        |-calconn---calconn+
                                                   |        |-replica---replica+
                                                   |        |-router---router--+
                                                   |        |-sched---sched---3+
                                                   |        `-update---update--+
                                                   |-server-+-Y4628185.sh
                                                   |        `-Y8609381.sh---ios+
                                                   `-server-+-cldbdir---cldbdir+
                                                            `-clrepl---clrepl--+

bash---pstree
```

*Figure 9-3 Output of the pstree command*

## ipcs

The **ipcs** command provides information about shared memory and semaphores; following are the options:

**-m** shared memory segments
**-q** message queues
**-s** semaphore arrays
**-a** all (this is the default)

You can use the **ipcs** command to check whether there are still any shared memory segments after an abnormal Domino server shutdown. All the shared memory segments for Domino begin with 0xf8. To display them, you can issue the command **ipcs -m|grep 0xf8** as shown in Figure 9-4.

```
ipcs -m|grep 0xf8
domserva@linuxa:~> ipcs -m|grep 0xf8
0xf8232000 1703936    domserva  660        5024752    13
0xf8232001 1736705    domserva  660        8388608    13
0xf8232002 1769474    domserva  660        262144     13
0xf8232003 1802243    domserva  660        262144     13
0xf8232004 1835012    domserva  660        262144     13
0xf8232005 1867781    domserva  660        262144     13
0xf8232006 1900550    domserva  660        262144     13
0xf8232007 1933319    domserva  660        262144     13
0xf8232008 1966088    domserva  660        262144     13
0xf8232009 1998857    domserva  660        262144     13
0xf823200a 2031626    domserva  660        8388608    13
```

*Figure 9-4 ipcs -m*

You can then use the **ipcrm** command to remove the shared memory IDs.

## 9.3.2 Performance monitoring with RMF PM for Linux

Resource Measurement Facility Performance Monitoring (RMF PM) with support for Linux Enterprise Server (RMF PMS) is a tool that can be used to monitor the various resources in a Linux on zSeries system. It is used, in conjunction with the RMF PM client application, to gather and analyze data.

Using RMF PM, you can:

1. Gather historical performance data.
2. Store the data in spreadsheet format.
3. Use a graphical user client to access performance data.
4. Generate graphical trend reports.

RMF PMS for Linux is a part of a larger family of RMF products. There is an RMF PM tool for z/OS; the same workstation client tool can be used to analyze z/OS and Linux systems.

At the time of writing, the RMF PM application for Linux was a Technology Edition. Both the server data gatherer portion of RMF PMS and the RMF PM client can be downloaded from the IBM RMF Web site. In addition to the code, there is a brief tutorial available at:

http://www.ibm.com/servers/eserver/zseries/zos/rmf/rmfhtmls/pmweb/pmlin.htm

Figure 9-5 presents a schematic overview of RMF PMS.

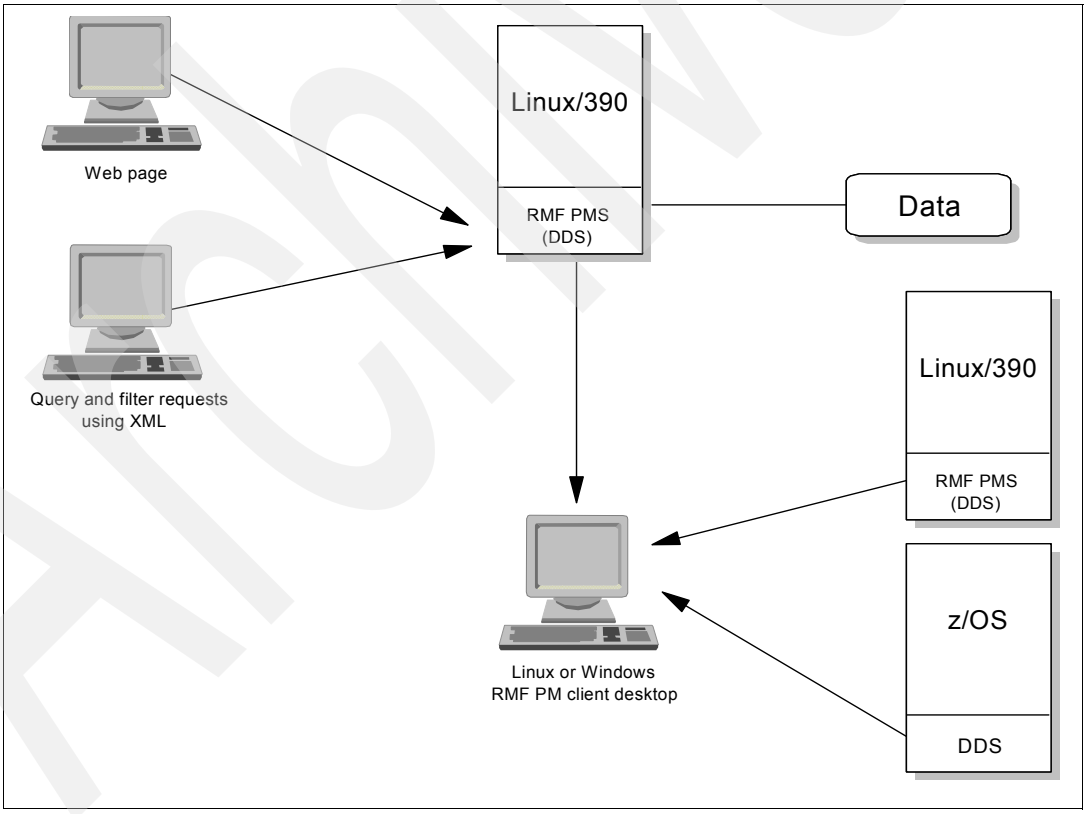



*Figure 9-5 RMF PMS overview*

### Gathering data on the Linux server

RMF PMS is the modular data gatherer portion of the tool, and it runs on the Linux server. The data gatherer modules work as daemons. There are daemons to gather data for CPU, network, filesystem, DASD I/O, memory, and the Apache HTTP server.

The daemons do not need to be synchronized; they can all be started or just selected ones can be started. The default interval collection time is 60 seconds.

When the data gatherer daemons have been started, data is collected in a directory on the Linux server. If the data gatherers are allowed to run continuously, RMF PMS will create a new directory at midnight to hold the data for that day. You can extract this data to an application like the RMF PM workstation, or to another application that you develop. RMF PMS provides an archive function to reduce the amount of active data on your Linux server.

The following lists present a few examples of the available metrics. For more information, see the online documents in the rmfpms directory.

#### *CPU resource metrics*

- Load average
- Percent of CPU total active by processor
- Percent of CPU idle time
- Percent of CPU time in kernel mode by process
- Accumulated CPU time in user mode by process

#### *Network resource metrics*

- Bytes received/transmitted
- Packets received/transmitted by network device
- Receive/transmit errors

#### *Filesystem resource metrics*

- Space available
- Size of all filesystems
- DASD I/O requests per second
- Percent of space used
- DASD I/O average response time per request

#### *Memory resource metrics*

- Memory used
- Swap space used
- Cache memory
- Number of pages swapped in/out
- Shared memory
- Total memory size

### Analyzing the data using the RMF PM client

The option we used for analyzing the data was the RMF PM client. The client is available for Linux and the Windows desktop. We used the Windows desktop.

The RMF PM client connects to the Linux server and extracts the data which has been gathered. The client can extract the data in real time, sampling at 60-second intervals, or can extract historical data which has been gathered over a longer period of time (a day, for example).

The initial client desktop is shown in Figure 9-6 on page 234.

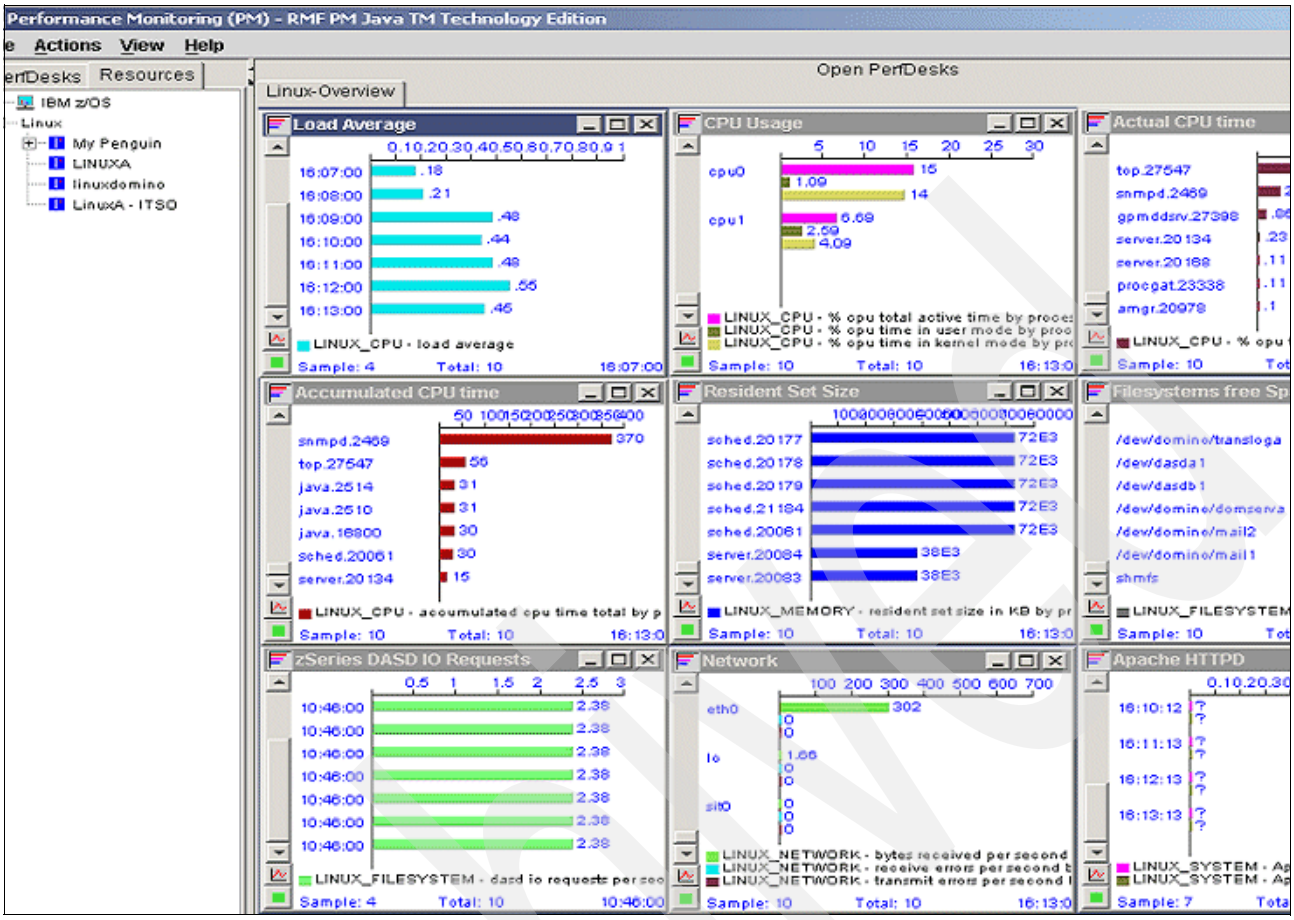



*Figure 9-6 RMF PM client desktop*

Both the data gathering on the server and the sampling from the PM client can consume large amounts of CPU if you use the default server and client setup. We recommend that you decide what data you need to gather and disable the other data gatherer daemons.

For example, if you are interested primarily in CPU utilization, you may not need to gather data on the filesystem or I/O requests, so you can disable the file and DASD daemons. If you are not running the Apache HTTP server, you don't need to start the daemon. By the same token, if you do not need to analyze all the data at the workstation, you can delete some of the DataViews from your performance desktop.

Data at the client is presented in DataViews. Each of the small windows shown in Figure 9-6 is a DataView. Initially the client captures data for nine DataViews. We did not have the Apache HTTP server, so that DataView is empty.

You can customize which DataViews you have on your performance desktop, and you can selectively capture and analyze data for each one. The DataView windows can be resized and maximized to provide more information. So, you are not limited to seeing the small number of processes shown in the Actual CPU time DataView in the figure. If you maximize the window, you will see all the running processes.

The DataViews that you see on the desktop provide a snapshot of the Linux server at one sampling. You can scroll forward and backward through the samples. You can also take a series of these samples and graph them. Figure 9-7 on page 235 shows a graph of 13 samples of CPU utilization for the Domino Server task taken August 27 from 16:04 to 16:16.

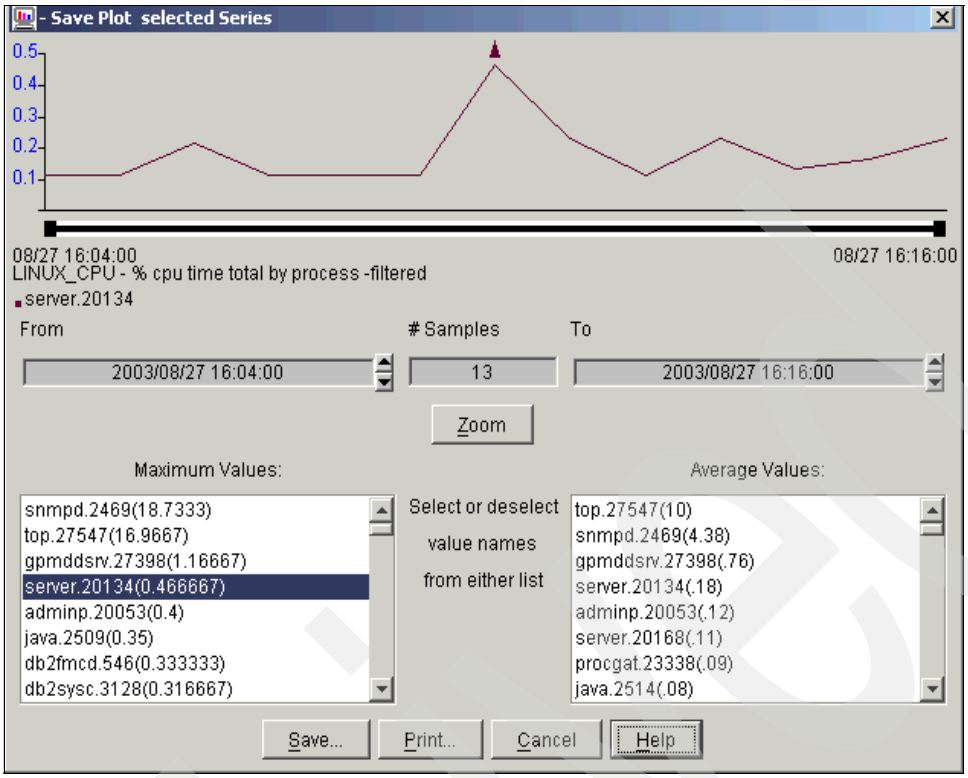



*Figure 9-7 RFM PM graph*

## 9.4 VM administration

When you run Linux under z/VM, there are times when you must simulate real processor or hardware functions. Use the following CP commands to simulate these real operator functions. Remember, to use any CP commands, you must precede them with #CP. Refer to *z/VM: CP Command and Utility Reference* for more details and commands.

### 9.4.1 VM CP commands

- Use the **ATTACH** command to attach the following:
  - Connect a real device (for example, OSA) to your virtual machine
  - Expanded Storage to your virtual machine

  **Note:** This is a privileged command.
- Use the **COUPLE** command to connect two virtual channel-to-channel adapters (CTCAs) or a Network Interface Card (NIC) to a Guest LAN.
- Use the **DEFINE** command to:
  - Change the memory size of your virtual machine
  - Create virtual CTCAs
  - Create temporary minidisks
  - Create additional virtual CPUs
  - Create Guest LANs

- Use the **DETACH** command to detach the following:
  - Virtual processors from your virtual machine
  - Virtual devices from your virtual machine
  - Expanded Storage from your virtual machine
  - Virtual message processor and message devices from your virtual machine
  - VM LAN segments
- Use the **DISPLAY** command to see data from both real and virtual storage.
- Use the **IPL** command to simulate an Initial Program Load (boot) for your virtual machine. “IPL” and” boot” are synonymous throughout this chapter.
- Use the **CP TERMINAL HOLD** command to control whether CP displays the Holding status when the terminal screen is full.
- Use the **CP TERMINAL MORE** command to change the number of seconds that elapse between the time when CP issues the MORE... state and sounds the terminal alarm before CP clears the screen.
- Use the **CP TRACE** command to monitor events that occur in your virtual machine.
- Use the **CP VMDUMP** command to dump all or selected pages from the virtual machine’s storage.

## 9.4.2 VM commands

There are various standard VM commands that you can use to display performance information for your system.

### Indicate command

The CP command **indicate** allows you to obtain information about the status of the z/VM guest system, as well as the status of the system resources of z/VM.

If you are a class G user, you can use **indicate** to display:

- Recent contention for system resources. This can be helpful to predict system throughput and response time characteristics that your virtual machine may experience.
- Environment characteristics of your virtual machine. This includes machine type, the origin of the system IPLed (loaded) in your virtual machine, and the presence or quantity of system resources available to your virtual machine.
- Measurements of resources used by your virtual machine. These measurements are *accumulators*, which means they are always increasing after the logon of your virtual machine.

If you are a class E user, the **indicate** command provides all class G functions and the following:

- Detailed information about use of, and contention for, system resources. User IDs of virtual machines currently using certain resources can be displayed.
- The status of current active virtual machines as determined by the system scheduler and dispatcher.
- Environment characteristics of, and measurements of resources used by, any virtual machine logged on.

Example 9-8 shows the output from the **indicate** command.

*Example 9-8 Output of the indicate command*

```
indicate
AVGPROC-008% 02
XSTORE-000000/SEC MIGRATE-0000/SEC
MDC READS-000003/SEC WRITES-000001/SEC HIT RATIO-095%
STORAGE-036% PAGING-0000/SEC STEAL-000%
Q0-00000(00000)                          DORMANT-00042
Q1-00000(00000)           E1-00000(00000)
Q2-00000(00000) EXPAN-001 E2-00000(00000)
Q3-00006(00000) EXPAN-001 E3-00000(00000)

PROC 0000-008%          PROC 0001-009%

LIMITED-00000
Ready; T=0.01/0.01 16:27:57
```

The **indicate active** command shows the total number of users active in a specified time interval (the default is 60 seconds), and the number of users in the dispatch, eligible, and dormant list that were active in the specified time interval

```
indicate active
0023 USERS, 0006 DISP, 0000 ELIG, 0017 DORM
Ready; T=0.01/0.01 16:29:50
```

The **indicate user** command displays performance-related information for a Linux virtual machine. This command can be issued from the console of a running Linux virtual machine by prefacing the command with the current terminal line end character (normally a # character); for example, **#CP indicate user**.

Example 9-9 shows the output from the **indicate user linuxc** command issued from user maint.

*Example 9-9 indicate user command*

```
indicate user linuxc
USERID=LINUXC   MACH=XA  STOR=392M VIRT=V XSTORE=NONE
IPLSYS=DEV 0200 DEVNUM=00051
PAGES: RES=00091285 WS=00088431 LOCK=00002630 RESVD=00000000
NPREF=00000000 PREF=00000000 READS=00000001 WRITES=00000001
XSTORE=000000 READS=000000 WRITES=000000 MIGRATES=000000
CPU 00: CTIME=21:44 VTIME=009:12 TTIME=010:03 IO=044684
        RDR=000000 PRT=000000 PCH=000000

USERID=LINUXC   MACH=XA  STOR=392M VIRT=V XSTORE=NONE
IPLSYS=DEV NONE DEVNUM=00051
PAGES: RES=00091285 WS=00000000 LOCK=00002630 RESVD=00000000
NPREF=00000000 PREF=00000000 READS=00000001 WRITES=00000001
XSTORE=000000 READS=000000 WRITES=000000 MIGRATES=000000
CPU 01: CTIME=21:44 VTIME=008:31 TTIME=009:22 IO=016957
        RDR=000000 PRT=000000 PCH=000000
Ready; T=0.01/0.01 16:24:21
```

This response gives all data from the user's VMDBK relevant to the user's virtual machine paging activity, resource occupancy, processor usage, and accumulated I/O activity counts since logon. Time and count values are ever-increasing accumulators.

Example 9-10 on page 238 shows output from the **indicate I/O** command.

*Example 9-10 Output from indicate i/o command*

```
indicate i/o
LINUXB   1518 LINUXB   ---- LINUXC   1519 LINUXC   ---- LINUXA   3A43
TCPIP    ---- RSCS     ---- PVM      ----
Ready; T=0.01/0.01 16:32:43
indicate paging
No users in page wait
Ready; T=0.01/0.01 16:33:52
indicate load
AVGPROC-007% 02
XSTORE-000000/SEC MIGRATE-0000/SEC
MDC READS-000003/SEC WRITES-000001/SEC HIT RATIO-094%
STORAGE-036% PAGING-0001/SEC STEAL-000%
Q0-00000(00000)                           DORMANT-00043
Q1-00000(00000)           E1-00000(00000)
Q2-00000(00000) EXPAN-001 E2-00000(00000)
Q3-00006(00000) EXPAN-001 E3-00000(00000)

PROC 0000-007%          PROC 0001-008%
LIMITED-00000
```

## VM query commands

You can use the Q DASD and the Q USER commands to display dasd and users, as shown in Example 9-11.

*Example 9-11 q dasd & q user output*

```
Q dasd
DASD 1518 CP SYSTEM LX1518   1
DASD 1519 CP SYSTEM LX1519   1
DASD 151A CP SYSTEM LX151A   1
DASD 151B CP SYSTEM LX151B   1
DASD 151C CP SYSTEM LX151C   1
DASD 151D CP SYSTEM LX151D   1
DASD 151E CP SYSTEM LX151E   1
DASD 151F CP SYSTEM LX151F   1
DASD 1558 CP SYSTEM LX1558   1
DASD 1559 CP SYSTEM LX1559   1
DASD 155A CP SYSTEM LX155A   1
DASD 155B CP SYSTEM LX155B   1
```

**q user**

```
   46 USERS,     0 DIALED,     0 NET
Ready; T=0.01/0.01 16:37:48
```

You can see whether mdcache is on, as follows:

```
q mdcache
Minidisk cache ON for system
Storage MDC min=0M max=2047M, usage=37%, bias=1.00
Xstore MDC min=0M max=2048M, usage=95%, bias=1.00
Ready; T=0.01/0.01 16:40:46
```

You can display the minidisks that belong to a user ID, as shown in Example 9-12 on page 239.

*Example 9-12 q mdisk*

```
q mdisk userid linuxc 191-310 det
TargetID Tdev OwnerID  Odev Minidisk  DEVNO Duplex
LINUXC   0191 LINUXC   0191 Regular   No    No
LINUXC   019D MAINT    019D Regular   No    No
LINUXC   019E MAINT    019E Regular   No    No
LINUXC   019F MAINT    019F Regular   No    No
LINUXC   0200 LINUXC   0200 Regular   No    No
LINUXC   0201 LINUXC   0201 Regular   No    No
LINUXC   0202 LINUXC   0202 Regular   No    No
LINUXC   0203 LINUXC   0203 Regular   No    No
LINUXC   0204 LINUXC   0204 Regular   No    No
```

## 9.5 Automating Domino startup and shutdown

The Domino server, either running under Linux natively in an LPAR or running as a guest operating system under VM, presents an unusual problem. It needs to be shut down properly and completely before the Linux operating system is shut down. If Linux is shut down without Domino closing all databases, corruption will likely occur in one or more databases.

The problem is a bit more complex when Domino is running in a Linux virtual machine. In this case, Domino is two levels down from the base operating system, z/VM.

Ideally, when the VM system is IPLed, we want to have both the Linux guest operating systems and the Domino servers started. Additionally, when it is necessary to shut down the VM system for maintenance or other reasons, we need to insure that the Domino server is safely shut down before shutting down the Linux guests and VM operating system. Abnormal shutdown of the Domino server is a prime cause of database corruption.

To solve these problems, automating the startup and shutdown of the Linux guests and the Domino servers is made possible by using a mixture of VM and Linux command functions.

### 9.5.1 Automated startup

Starting and stopping a Domino server when Linux is IPLed and shut down is accomplished by executing a script which resides in the /etc/init.d directory. This script is enabled from the **chkconfig** command.

If Domino servers are present, multiple copies of the script can be enabled and executed to safely manage the server. The steps for setting up the server for automatic start and stop will work regardless of whether the Domino server is being run under Linux in an LPAR or as a VM guest.

#### Setup required at the Linux level

The script shown in Example 9-13 on page 240 will start the Domino server when Linux is logged on, and shut down the Domino server when a Linux shutdown has been requested. On shutdown, it will wait until the server or servers have shut down before allowing Linux to shutdown.

For the sake of clarity, we refer to this as the domservb script for the remainder of the section (you can use any name you want for the script). Our personal preference was to use the actual name of the server (that is, domservb), as this allows for multiple copies of the script to be used to start multiple DPARs.

For the script to work, the .profile for the server must contain path statements for the location of the notes.ini file and the server:

```
export DOMINO_LINUX_SET_PARMS=1
export PATH=/usr/local/bin:/usr/bin:/bin:/opt/lotus/bin:/domservb/notesdata
```

The following variables need to be modified in the script to allow it to work properly:

```
DOMINO_BIN=/opt/lotus/bin/server
DOMINO_USR=domservb
```

This script was developed and tested as part of the redbook project. It can be freely modified to suit your needs. While it worked in our environment, it may need modification to work correctly in a different environment. (No warranty is expressed or implied.) Check the release notes for your version of Domino for any information about starting or stopping Domino that might be provided.

**Important:** The automated startup script (domservb) requires root authority to run.

*Example 9-13 Sample automated startup/shutdown script*

```
#!/bin/sh
#
# /etc/init.d/domino
#
### BEGIN INIT INFO
# Provides: domino
# Required-Start: $network $remote_fs
# Required-Stop: $network
# Default-Start: 3 5
# Default-Stop: 0 1 2 6
# Description: Domino server
### END INIT INFO

. /etc/rc.status

# set the domino path and user name under which domino will run
DOMINO_BIN=/opt/lotus/bin/server
test -x $DOMINO_BIN || exit 5

DOMINO_USR=domservc
DOMINO_TESTS=/opt/lotus/notes/latest/zlinux/server
DOMINO_TESTC=/opt/lotus/notes/latest/zlinux/jvm/bin/exe/java

rc_reset

case "$1" in
    start)
        if checkproc $DOMINO_TESTS; then
                echo -n "Domino server is already running."
                rc_status -v
                exit
        fi
        if checkproc $DOMINO_TESTC; then
                echo -n "Domino Server Controller is already running."
                rc_status -v
                exit
        fi
        if test -e /$DOMINO_USR/notesdata/.jsc_lock;then
                rm /$DOMINO_USR/notesdata/.jsc_lock
```

```
                echo -n "Controller was not shutdown properly. .jcs_lock file removed. "
        fi
   echo 131072 > /proc/sys/fs/file-max
   echo 15 > /proc/sys/net/ipv4/tcp_fin_timeout
   echo 16384 > /proc/sys/net/ipv4/tcp_max_syn_backlog
   echo 1 > /proc/sys/net/ipv4/tcp_tw_reuse
   echo "1024    65535" > /proc/sys/net/ipv4/ip_local_port_range
        MAX_OPEN_FILES=`ulimit -n`
        if [ $MAX_OPEN_FILES -lt 20000 ]; then
            echo "setting maximum open files to 20000"
            ulimit -n 20000
        fi
   echo -n "Starting Domino server"
   echo "$DOMINO_BIN -jc -c &" | su - $DOMINO_USR
   rc_status -v
   echo
   ;;
    stop)
   echo -n "Shutting down Domino server"
   echo "Y" > /tmp/domino_$$
   su - -c "$DOMINO_BIN -jc -q"  $DOMINO_USR < /tmp/domino_$$
   rm /tmp/domino_$$
   rc_status -v
   echo
   ;;
    restart)
        $0 stop
        $0 start
        rc_status
        ;;
    status)
        echo -n "Checking for Domino server: "
        checkproc $DOMINO_TESTS
        rc_status -v
      echo
        ;;
    *)
   echo "Usage: $0 {start|stop|restart|status}"
   exit 1
   ;;
esac

rc_exit
```

To use this script, create a file called domservb. Type in the script and save it into the /etc/init.d directory. It must be placed into this directory in order to work.

> **Important:** When setting up the script, you should be working in the /etc/init.d directory. When issuing the **domservb** command from the /etc/init.d directory, you should use the **./domservb** form of the command. To issue this command from any other directory when logged on as root or superuser, you must have /etc/init.d in the path.

Use the **chmod** command to correctly set the execute (x) permission bits on the file.

```
linuxc:/etc/init.d # ls -al dom*
-rw-r--r--    1 root     root         1588 Aug 26 17:47 domservb

linuxc:/etc/init.d # chmod 755 domservb

linuxc:/etc/init.d # ls -al dom*
-rwxr-xr-x    1 root     root         1588 Aug 26 17:47 domservb
```

Use the **chkconfig** command to enable the script:

```
linuxc:/etc/init.d # chkconfig domservb on
```

With this command entered, the domservb script will start the Domino server and the Java-based remote console controller on Linux startup, and shut down the server and controller on Linux shutdown.

Use the **chkconfig** command to disable the script:

```
linuxc:/etc/init.d # chkconfig domservb off
```

You can issue the **chkconfig** command without the on or off parameter to check the status of the script:

```
chkconfig domservb
```

The script accepts a number of commands which can be used to control the server:

| | |
|---|---|
| **stop** | Stops the server and the remote Java-based remote console controller |
| **start** | Starts the server and the Java-based remote console controller |
| **restart** | Stops an restarts the server and the Java-based remote console controller |
| **status** | Returns the status of the server |

To check whether the Domino script works properly, issue the following command with the server down:

```
domservb start
```

With the server start verified through the Domino remote console, test server shutdown:

```
domservb stop
```

Status of a server can also be checked:

```
domservb status
```

Possible responses are:

```
Checking for Domino server:                                    running
Checking for Domino server:                                    unused
```

These show whether the server is running or unused (down).

## How to run multiple DPARs using the script

Multiple servers can be stopped and started automatically simply by renaming the scripts to the server's name, modifying the `DOMINO_USR=` variable in the script to the server's name, placing the scripts in the /etc/init.d directory, and enabling the scripts with **chkconfig**.

### Automating Server shutdown

The same domservb script that starts the server also stops the server when the Linux **shutdown** command is entered. The script issues a quit to the Domino server and controller, then waits until the Domino server shutdown is complete before returning control to the Linux shutdown process. No interaction is needed other than to issue the **shutdown** command to Linux. If multiple DPARs are running within the Linux image, and if they have been started using the automation script, they too will be shut down before Linux terminates.

## 9.5.2 Automated startup on IPL of the VM operating system

When the z/VM operating system is IPLed, a virtual service machine is automatically logged under the user ID AUTOLOG1; this is the default for z/VM. The PROFILE EXEC contains CP XAUTOLOG statements which automatically log on virtual service machines and virtual machines running guest operating systems like Linux.

When the guest virtual machines are logged on, the guest's PROFILE EXEC is executed if IPL CMS PARM AUTOCR has been specified in the VM directory entry for the user. The PROFILE EXEC contains commands which can be conditionally executed to set up the environment and IPL the guest OS.

Upon IPLing Linux, various start/shutdown scripts can be configured to run. We can add a script to Linux to start up a Domino server or multiple servers; see 9.5.1, "Automated startup" on page 239. Using a combination of XAUTOLOG, PROFILE EXEC and a Domino startup script in Linux, we can automate the startup of Domino.

If AUTOLOG1 is not used to logon the Linux guests, a REXX EXEC can be run from the VM OPERATOR user ID, which is typically the ID that is running on the VM console. This REXX EXEC would contain the CP XAUTOLOG statements for the Linux guests.

### Setup required at the VM level

Add a line to the PROFILE EXEC on the AUTOLOG1 191 minidisk for each of the LINUX guests to be automatically started:

```
CP XAUTOLOG LINUXA
CP XAUTOLOG LINUXB
CP XAUTOLOG LINUXC
```

This would cause LINUXA, LINUX B, and LINUXC to be autologged at IPL time.

The VM directory entries for each of the Linux guests running Domino servers must have the following statement:

```
IPL CMS PARM AUTOCR
```

Each of the Linux guests running Domino servers must have a PROFILE EXEC that determines whether the guest virtual machine has been started in disconnected state, and if so, IPL the Linux OS. Additional setup tasks can be done as shown in Example 9-14:

*Example 9-14 Sample profile exec*

```
/* Profile Exec
   'ACCESS 19F Z'                      /* Access local tools disk.  */
   'CP SET PF11 RETRIEVE FORWARD'      /* Retrieve first command     */
   'CP SET PF12 RETRIEVE'              /* Retrieve last command */
   'CP SET PF23 RETRIEVE FORWARD'      /* Retrieve first command  */
   'CP SET PF24 RETRIEVE'              /* Retrieve last command */
/* If the ID has been autologged in disconnected state, start linux /*
/* Else exit to CMS.
```

```
if word(diagrc(24,-1),2)=2 then       /* Is the user disconnected */
'exec swapgen 301 200000 (DIAG'        /* Setup VDisks for swapping */
'exec swapgen 302 200000 (DIAG'
 'exec swapgen 303 200000 (DIAG'
 'exec swapgen 304 200000 (DIAG'
 'exec swapgen 305 200000 (DIAG'
 'exec swapgen 306 200000 (DIAG'
 'exec swapgen 307 200000 (DIAG'
 'exec swapgen 308 200000 (DIAG'
 'exec swapgen 309 200000 (DIAG'
 'exec swapgen 310 200000 (DIAG'
 'cp set run on'
 'CP IPL 200 clear'                    /* IPL Linux                */
 /* Exit
```

This PROFILE EXEC, which we used in our redbook project, will determine if the guest has been started in disconnected state (autologged). If so, it will use the SWAPGEN exec to prepare the virtual disks for use as Linux swap, and then IPL Linux (IPL 200). Upon IPL of the Linux OS, the startup script for the Domino server is called and the Domino server is started automatically as part of Linux startup.

### 9.5.3 Automated shutdown when running Linux under VM

Linux guests can be shut down from VM using a REXX EXEC started at the VM operator console. The REXX EXEC will use two CP commands to accomplish the passing of the shutdown command to the LINUX guests.

**CP SEND** This CP command sends console input through VM to a guest virtual machine. This guest (Linux, in our case) acts on the command as though it had been issued by a logged on user. To use CP send, a user must previously set the **CP SET SECUSER** command. This command requires a class C user privilege.

**CP SET SECUSER** This command is used to set the issuing ID as a secondary user for a specified target virtual machine (in our case, a Linux guest). Until it is reset, the issuing ID will also receive the console responses from the target virtual machines console. This command requires the issuing ID be privilege class C.

The Linux **shutdown** command must be submitted by someone with root authority. Under normal circumstances, commands sent to the Linux guests using the **CP SEND** command will require the use of a password. There are two methods of dealing with this requirement:

- Append a root password to the **CP SEND** command sent from class C operator ID.
- Open a root console login on the Linux guest.

The first option is undesirable, since it exposes the critical root password to any user with access to the STPLINUX exec. Although the exec could be crafted in such a way as to require that the operator supply a password when the STPLINUX exec is run, the better approach is to create a shell script to log on root when Linux is initialized.

To enable root as login on Linux initial load:

1. Create the /root/bin/consolescript file:

   ```
   #!/bin/sh
   exec /bin/login -f root
   ```

2. Set the execute permissions for the script:

```
chmod 0700 /root/bin/consolescript
```

3. Make a copy of /etc/inittab:

```
cp -p /etc/inittab inittab.bkup
```

4. Edit /etc/inittab.

   Replace the following line:

```
1:1235:respawn:/sbin/mingetty console
```

   With this line:

```
1:1235:respawn:/sbin/agetty -L -n -l /root/bin/consolescript 9600 console dumb
```

Set up the STPLINUX EXEC on an ID with sufficient privilege to issue **CP SEND** and **CP SET SECUSER**. The OPERATOR ID is recommended as the proper place for this exec since it will be used to shut down the VM system.

The STPLINUX exec should be incorporated into the execs that are used to shut down service virtual machines prior to VM shutdown. It is further recommended that it be one of the first execs executed in order to give the Linux guest sufficient time to shut down properly.

```
/* stplnx - stops linux  -                   */

 trace all
 'Pipe cms id () | spec w1 | var user'    /*Set user to current user */
 say ''
 say ' Shutting down the Linux systems '
 say ''

 /* list Linux system(s) in variable linuxid                        */
 /* If multiple systems list them all separated by comma            */
 /* List user ID of the system issuing this exec in variable user     */

 user = maint
 linuxid = linuxc linuxb
 'pipe',
    ' literal 'linuxid,
    '| split',
    '| stem linuxid.'

do i=1 to linuxid.0
 /* Set secuser, issue command, and reset secuser                    */

   'SET SECUSER 'linuxid.i user
 address command 'CP SEND 'linuxid.i 'shutdown -t -h 1 now'
   'SLEEP 10 SEC'

'SET SECUSER 'linuxid.i' RESET'
 end

 Exit
```

When the STPLINUX exec is executed, it will issue the **shutdown** command to Linux. Linux will then shut down the Domino servers that are running in the guest. If there are multiple Linux guests to shut down, enter the user ID of each Linux guest in the variable linuxid, separated by a comma.

### 9.5.4 Operating system maintenance with CheckOS

CheckOS is a script used to verify that the operating system contains the appropriate patch level in order to run Domino 6. The script is installed during Domino installation and resides in the Domino binaries directory (/opt/lotus/bin/checkos).

This script will run when you install Linux. You will see a couple of lines, including a link to the latest patches, and a line of information while the tool gathers the data. Next, the script checks and reports which OS you are running on the system, followed by the machine type and filesets required for the Domino 6 server to run properly. If there are any files missing, they will be reported in the section “The following OS patches are required:” You need to install the missing patches before continuing.

**Note:** At the time of writing, CheckOS was not used by Domino for Linux on zSeries. Check the Release Notes with your Domino version for the latest information about its capabilities.

## 9.6 Problem determination

When there are problems on the Domino server on Linux, the Domino Administrator and the Linux Administrator will need to work closely to determine the root cause of the problem and resolve it.

### 9.6.1 Domino server problems

If the Domino server has crashed, there may have been a core dump generated. You will need to plan with the Domino Administrator in which directory to place the core dumps. By default, it will be generated in the Domino data directory, but the Domino Administrator can change it to a different location by a parameter in the notes.ini file (this is described in Chapter 8, “Domino administration” on page 173 of this book).

You will need to allocate sufficient space for the dumps. It is also advisable to compress core dumps using gzip to save disk space. You should also clean up this filesystem on a regular basis.

At times, the Domino administrator will need to contact the Linux administrator to check on the status of resources. For example, in 8.6, “Problem determination” on page 208, the Domino Administrator has been alerted to a mail backlog condition and has checked and ruled out any Domino server issues (server availability, corrupt file, and so on). So the Domino administrator would contact the Linux administrator.

At this point, one of the items that the Linux administrator can check is the DASD I/O. This can be done by using the `iostat` command. The `iostat` command reports the CPU utilization and the device utilization.

We ran iostat with the `-x` option to display extended statistics, as shown in Figure 9-8 on page 247. The first section reports CPU utilization on the user and system level, as well as percentage of time that the CPU or CPUs were idle. In the device section, the following stats are important to check:

**await** Average time (in milliseconds) for I/O requests issued to the device

**svctm** Average service time (in milliseconds) for I/O requests issued to the device

**%util** Percentage of CPU time during which I/O requests were issued to the device

```
domserva@linuxa:/domserva/notesdata > iostat -x 40 2
Linux 2.4.19-4suse-SMP (linuxa)         08/28/03

avg-cpu:  %user   %nice    %sys   %idle
           1.97    0.00    2.23   95.80

    Device:    rrqm/s wrqm/s   r/s   w/s  rsec/s  wsec/s    rkB/s    wkB/s avgrq-sz avgqu-sz   await  svctm
    %util
    /dev/dasda   0.90   2.38  0.48  0.57   11.00   23.55     5.50    11.78    33.19     0.17  165.58  46.85
    0.49
    /dev/dasda1
                 0.90   2.38  0.48  0.57   11.00   23.55     5.50    11.78    33.19     0.17  165.59  46.85
    0.49
    /dev/dasdb   1.77   0.76  0.45  0.23   17.78    7.91     8.89     3.96    37.97     0.09  136.48  30.11
    0.20
    /dev/dasdb1
                 1.77   0.76  0.45  0.23   17.78    7.91     8.89     3.96    37.97     0.09  136.50  30.11
    0.20
    /dev/dasdc   1.53   0.00  0.40  0.01   15.41    0.09     7.71     0.04    37.85     0.02   45.02  34.91
    0.14
    /dev/dasdc1
                 1.53   0.00  0.40  0.01   15.41    0.09     7.71     0.04    37.86     0.02   45.02  34.91
    0.14
    /dev/dasdd   1.50   0.00  0.41  0.01   15.26    0.06     7.63     0.03    37.00     0.02   45.43  35.42
    0.15
    /dev/dasdd1
                 1.50   0.00  0.41  0.01   15.26    0.06     7.63     0.03    37.00     0.02   45.43  35.42
    0.15
/dev/dasde   1.54   0.00  0.41  0.01   15.63    0.05     7.82     0.02    37.42     0.01   22.24  17.98   0.08
```

*Figure 9-8 iostat -x command*

## Domino and Linux administrators working on a problem

The Domino Administrator has the ability to collect platform statistics with the Domino command **Show Stat Platform**, and the Linux Administrator can use OS commands to obtain information from the system using cat /proc/meminfo and **iostat**.

Example 9-15 checks the CPU using **iostat -c** command and system output from platform statistics.

*Example 9-15 Checking CPU*

### Linux

```
domserva@linuxa:/domserva/notesdata > iostat -c
Linux 2.4.19-4suse-SMP (linuxa)         09/02/03

avg-cpu:  %user   %nice    %sys   %idle
           2.32    0.03    2.73   94.92Domino:
```

### Domino

```
Platform.System.PctCombinedCpuUtil = 0
  Platform.System.PctCombinedCpuUtil.Avg = 11.26
  Platform.System.PctCombinedCpuUtil.Peak = 97
  Platform.System.PctTotalPrivilegedCpuUtil = 0
  Platform.System.PctTotalPrivilegedCpuUtil.Avg = 6.1
  Platform.System.PctTotalPrivilegedCpuUtil.Peak = 64
  Platform.System.PctTotalUserCpuUtil = 0
  Platform.System.PctTotalUserCpuUtil.Avg = 5.02
  Platform.System.PctTotalUserCpuUtil.Peak = 52
  Platform.Time.LastSample = 09/02/2003 20:29:40 EDT
  Platform.Time.SampleRateInMins = 1
```

Example 9-16 checks memory using cat /proc/meminfo and memory output from platform statistics. (Notice that what Domino reports as paging statistics, Linux reports as swapping statistics, but it is the same information.)

*Example 9-16 Checking memory*

**Linux**

domserva@linuxa:/domserva/notesdata > cat /proc/meminfo

```
        total:    used:    free:  shared: buffers:  cached:
Mem:  195035136 192409600  2625536        0 32403456 110645248
Swap: 507617280 139149312 368467968
MemTotal:       190464 kB
MemFree:          2564 kB
MemShared:           0 kB
Buffers:         31644 kB
Cached:          96028 kB
SwapCached:      12024 kB
Active:          26488 kB
Inactive:       127504 kB
HighTotal:           0 kB
HighFree:            0 kB
LowTotal:       190464 kB
LowFree:          2564 kB
SwapTotal:      495720 kB
SwapFree:       359832 kB
```

**Domino**

```
Platform.Memory.PagesPerSec = 0
  Platform.Memory.RAM.AvailMBytes = 4
  Platform.Memory.RAM.AvailMBytes.Avg = 1
  Platform.Memory.RAM.AvailMBytes.Min = 1
  Platform.Memory.RAM.AvailMBytes.Peak = 11
  Platform.Memory.RAM.PctUtil = 97
  Platform.Memory.RAM.TotalMBytes = 186
  Platform.PagingFile.Free.SizeMBytes = 351
  Platform.PagingFile.Total.PctUtil = 27
  Platform.PagingFile.Total.PctUtil.Avg = 19
  Platform.PagingFile.Total.PctUtil.Peak = 34
  Platform.PagingFile.Total.SizeMBytes = 484
```

Example 9-17 on page 249 checks iostats using the **`iostat -d`** command and logical output from platform statistics.

*Example 9-17 Checking iostats*

**Linux**

```
domserva@linuxa:/domserva/notesdata > iostat -d -x dasdl
Linux 2.4.19-4suse-SMP (linuxa)         09/02/03

Device:    rrqm/s wrqm/s   r/s   w/s  rsec/s  wsec/s    rkB/s    wkB/s avgrq-sz
avgqu-sz   await  svctm  %util
/dev/dasdl   0.98   0.01  0.24  0.01    9.78    0.16     4.89     0.08    39.84
    0.01   29.71  21.37   0.05
```

**Domino**

```
Platform.LogicalDisk.10.AssignedName = dasdl
  Platform.LogicalDisk.10.AvgQueueLen = 0
  Platform.LogicalDisk.10.AvgQueueLen.Avg = 0.05
  Platform.LogicalDisk.10.AvgQueueLen.Peak = 0.91
  Platform.LogicalDisk.10.PctUtil = 0
  Platform.LogicalDisk.10.PctUtil.Avg = 0.13
  Platform.LogicalDisk.10.PctUtil.Peak = 7.55
  Platform.LogicalDisk.10.ServiceTime = 0
  Platform.LogicalDisk.10.ServiceTime.Avg = 11.18
  Platform.LogicalDisk.10.ServiceTime.Peak = 100
```

Using the two output commands will show both Domino and Linux Administrators where bottlenecks may be occurring and what means are needed to correct the situation.

## 9.7 Backup/recovery

When running Lotus Domino, like any application on zSeries, it is necessary to have a well-defined backup strategy in place to ensure data recovery. This section lists some options for Domino. We do not document detailed implementation instructions for these options, but we present a high level point of view, listing advantages and disadvantages, as appropriate.

### 9.7.1 Understanding your backup and recovery needs

There are two basic categories of data in a Domino environment:

- Files, including system configuration data and the Domino executables, notes.ini, and ID files
- Domino databases and templates

We make this distinction because several methods and tools can be used to back up the files and Domino databases, including Linux archive commands such as tar and the TSM Backup-archive client. But these methods cannot be safely used to back up the Domino databases while the Domino server is active.

In order to take a good clean backup of Domino databases while the server is up, you need a utility which uses the Domino backup and recovery APIs. Although there are several products available for other platforms, at the time of writing, the only tool available for Linux on zSeries was the IBM Tivoli Storage Manager for Mail: Data Protection for Lotus Domino. We worked with a beta version of the Linux on zSeries client.

In regards to Domino data, there are three distinct recovery situations.

1. Full recovery: Common disaster recovery situations, like failure of a disk subsystem. All databases contained on the subsystem must be restored.
2. Database level recovery: A single database may have been corrupted. Recover by restoring the last backup of the database.
3. Document level recovery: One or more documents in a database have been lost, either deleted or corrupted. Normal recovery in this situation involves restoring the backup of the database to a temporary location and then copying the required documents to the active version of the database.

   Or you may be required to do a point-in-time restore (forward recovery from a full backup with application of transaction logs) of the database using the backup and recovery APIs in Domino.

The data on the Domino server can be backed up with different backup techniques. We review the following:

- The Backup-archive client of Tivoli Storage Manager (TSM)
- IBM Tivoli Storage Manager for Mail: Data Protection for Lotus Domino for Linux (formerly called TDP)

## 9.7.2 Tivoli Storage Manager (TSM)

Tivoli Storage Manager (TSM) is a client/server solution which provides backup, archive, and space management services to clients. Its functionality includes:

- Administrative clients that allow the administrator to control server activities, define storage management policies for workstation files, and set up schedules to provide automated backup and archive services at regular intervals.
- Support for backup archive clients that allow users to restore or retrieve files from a TSM server.
- Support for backing up and archiving files on a variety of devices and broad cross-platform and storage device support.
- Extensive and flexible centralized management that helps improve administrator productivity by enabling an administrator to proactively manage, operate on, and view many TSM servers from a single point.

These capabilities allow TSM to scale up to manage, administer, and automate data backup and restore. The base TSM provides two important functions:

1. Backup/Restore: IBM Tivoli Storage Manager includes multiple techniques to reduce data transfer sizes to their minimums to make backups and restores as fast as possible.
2. Archive/Retrieve: Tivoli Storage Manager can move data to offline storage to archive it and to free online disk space for more important active data. When you need that data again, IBM Tivoli Storage Manager can retrieve it for you.

Both the TSM Backup-archive client and the Tivoli Storage Manager Server Version 5.2 are available for Linux for zSeries systems. The server part of the solution (Tivoli Storage Manager server) can also run on other platforms such as AIX, z/OS, Windows, and other. You can get the latest information about the supported client and server platforms from:

http://www-3.ibm.com/software/tivoli/products/storage-mgr/platforms.html

At the time of writing, the latest release of TSM Backup-archive client available for Linux for zSeries was Version 5.2. In general, the TSM client version you use can be earlier or later than the TSM server version that you use. But you need to review the README that comes

with the TSM client to make sure you will have a compatible and interoperable client with your version of the TSM server.

### TSM Backup-archive client

The Backup-archive client has the ability to back up any online files, including Domino executables, databases and templates, notes.ini file, ID files, and Linux symbolic links.

There are two types of backup, incremental and selective:

- *Incremental* backs up files, directories, or subdirectories that are new or have changed since the last incremental backup.
- *Selective* backs up specific files or entire directories unconditionally.

To avoid fuzzy backups, specific backup options are set for the Backup-archive client in TSM's policies:

- Static: if being modified, TSM will not back up the file or directory.
- Shared static: if being modified, TSM will not back up the file or directory, but will retry the backup a predetermined number of times.
- Shared dynamic: if being modified, TSM will not initially back up the file or directory. It will retry the operation a predetermined number of times, and it will back up the file on its last attempt, even if it is being modified.
- Dynamic: TSM will always back up the file or directory, whether or not it is being modified.

For more information about TSM and the Backup-archive client, refer to *IBM Tivoli Storage Management Concepts,* SG24-4877.

Once the TSM client is installed, work with your TSM server administrator to create a Node and policy for your TSM client as well as to create the configuration files dsm.sys and dsm.opt. These contain details about what will be backed up and where to located the TSM server. They need to be placed in the /opt/tivoli/tsm/client/ba/bin directory. Example 9-18 shows a sample dsm.sys file.

*Example 9-18 Sample dsm.sys file for TSM backup/archive client*

```
TSM Client Configuration dsm.sys1
SErvername TSM_SERVER
COMMmethod TCPip
ERRORLOGName /var/log/tsmerror.log
ERRORLOGRetention 14D
MANAGEDServices webclient schedule
* NODename lnx19
PASSWORDAccess generate
SCHEDLOGName /var/log/tsmsched.log
SCHEDLOGRetention 14D
SCHEDMODe POlling
TCPPort 1500
TCPServeraddress tsm.itso.company.com
VIRTUALMountpoint /
exclude /var/log/tsm*.log
Example xx TSM Client Configuration dsm.opt
DATEformat 12
SUbdir Yes
```

Review your TSM Client documentation (or launch the TSM client interactively and type help) to get an explanation of the options.

### *Starting the TSM client*

TSM can back up your system interactively by executing the command: **`dsmc i`**.

The session will be displayed to your screen as the backup is processed, with a summary of the session upon completion. Alternatively, by starting the scheduler daemon (the most common way to use TSM), you can let the TSM server determine when the system is backed up. The simplest way of using the scheduler is by putting the following in your /etc/rc.d/rc.local script:

```
/opt/tivoli/tsm/client/ba/bin/dsmcad
```

The rc.local script is called when the system is booted. This line will ensure that the TSM scheduler client is started at that time. (This daemon automatically releases from the terminal, so there is no need to run this process in the background.) If the daemon is not running, you can start it by entering **`dsmc i`** while logged in as root.

> **Tip:** Try to keep your dsm.opt and dsm.sys files as generic as possible. This will enable you to distribute the same files to all your virtual Linux systems. When you have shared read-only filesystems, place the option files on that shared filesystem.

## 9.7.3 Data Protection for Lotus Domino

Data Protection for Lotus Domino is an application client that enables online backups and restores, as well as activation of Domino server database and transaction logs. IBM TSM for Mail: Data Protection for Lotus Domino is designed to use the backup and recovery API in Lotus Domino to provide online backup and restore capabilities using a Tivoli Storage Manager server. Unlike the TSM Backup-archive client, Data Protection always guarantees a consistent backup, since access to the data is through the Domino backup and recovery API.

Data Protection for Lotus Domino communicates with a Tivoli Storage Manager Server using the Tivoli Storage Manager API and it communicates with a Domino server using the Lotus Domino API.

Data Protection for Lotus Domino manages the Domino Server data and makes it easy to perform the following actions:

- Backup online Domino databases.
- Maintain multiple versions of Domino databases.
- Archive Domino transaction log files when archival logging is in effect.
- Restore backup versions of a Domino database and apply changes made since the backup from the transaction log.
- Restore Domino databases to a specific point in time.
- Expire database backups automatically based on version limit and retention period.
- Expire archived transaction log files when no longer needed.
- Automate scheduled backups.
- Data Protection for Lotus Domino provides two types of database backup (incremental and selective) and a log archive function.
- Incremental backup provides a full online backup of Domino databases when necessary. The specific conditions that determine when a new backup is necessary vary, depending on whether the database is logged or not.
- Selective backup unconditionally backs up the specified databases, unless they are excluded from backup through exclude statements.

- When archival logging is in effect, changes to logged databases can be captured in between full backups, by archiving the transaction log.

Even if you choose not to enable transaction logging, Data Protection for Lotus Domino can still be used to selectively and incrementally back up Domino databases and templates. It provides the only way to guarantee a consistent backup on an online Domino database, because it can access the pre-image data buffers through the Domino backup and recovery API. However, point-in-time recovery is not available without archive transaction logging enabled.

**Note:** Even when Data Protection for Lotus Domino is implemented, the TSM Backup-archive client is still needed for backing up the Domino executables, notes.ini file, and ID files.

Data Protection for Lotus Domino backs up only Domino databases and templates, including databases which are symbolically linked to the Domino data path by directory or database links.

## Restoring databases

There are two methods used to restore databases when transaction logs need to be applied. The database can be restored back to the same server it was on, usually into an alternate directory or with a different name. Or it can be restored to an alternate server specifically configured for the purpose of restoring databases.

### *Restoring files to the original server*

The option of restoring a file to the original server is the easiest method to implement, since no additional work is needed to set up this option. However, it is the method that has the most effect on performance, both for the server and for the restore process.

The process of restoring a database to the original server simply requires that the Data Protection for Lotus Domino restore command be run, followed by the **`activatedbs`** command. If needed, the archived transaction log extent files are restored to the active transaction log directory and replayed against the database.

There are two problems with this method. First, the transaction logging process performs best when the head is moving sequentially across the disk surface and not moving unnecessarily. Since the archived extent files must be written to the active directory, the head needs to be repositioned to an empty portion of the disk and the file written there. The log file must then be replayed to find all transactions that need to be applied to the database. In a restore, it may be necessary to restore and replay a large number of extent files.

Second, the transaction logging process will give absolute preference to writing new transactions to disk. The process of replaying the restored logs needs to wait for periods of lower activity to do its work. This results in long restore times.

Domino 6 introduced a new notes.ini parameter, TRANSLOG_RECOVER_PATH. It allows the path to which the archived transaction log extent files are restored to be specified. Since the restored extents no longer need to be restored to the active transaction log directory, this reduces disk I/O contention on the drive that contains the active transaction log. It allows the database to be restored more quickly if transaction logs need to be applied.

However, it can have an effect on overall server performance. Because the restore operation completes more quickly, it may also use more CPU in a concentrated period of time.

To use the alternate path for restoring transaction logging extent files, add the following line to the notes.ini of the Domino server:

```
TRANSLOG_RECOVER_PATH=<path to restore extents to>
```

#### *Restoring files to an alternate server*

The alternate method employed to restore databases is to set up a server, or rather a server shell structure, for restores. This server is never started as a Domino server; instead, it has directory tree structures mirroring those of the production server. It must be at the same Domino release level as the source server and must have a copy of the server.id and notes.ini files from the source server or servers.

Additionally, when a restore is requested, the most current transaction log extent files should be archived and restored to the restore server's transaction log directory. There is a twelve-step process, described in Appendix B of the Tivoli Data Protection for Lotus Domino manual, that needs to be followed to set up the restore server for restoring a database.

Most administrators use a series of scripts to set up the alternate server restore environment and process the restore. While the task of setting up this environment is a bit difficult, it offers the benefit of faster restores with no effect on the production Domino servers.

Since the archived extent files are restored to the transaction log directory of the restore server for replay, there is no effect on the production environment. This is the recommended method for restoring a database. Instructions for restoring to an alternate server or partition are contained in *IBM Tivoli Storage Manager for Mail: Data Protection for Lotus Domino for UNIX and OS/400 Installation and User's Guide*, SC32-9056.

### 9.7.4 Transaction logging

In 8.4, "Transaction logging" on page 202, we discuss how to enable transaction logging and the benefits of using it. There are additional benefits, with regard to backup and restore of Domino databases, offered by archive transaction logging:

- Reduced backup times: Backing up terabytes of data is a very time-consuming process, typically taking many hours to complete. When archived-style transaction logging is used, the amount of time needed to do daily backups of databases can be reduced by as much as 80 percent. This savings is realized because only the transactions which have been applied to databases are backed up on a daily basis, rather than the entire contents of the databases.

  The exact opposite is true for restores. The longer the interval between full backups, the longer the time to restore a database, since more logs must be processed.You should consider not only the backup times but also the restore times to determine the right balance of log versus full backups.

  Typically, a base full backup of all databases is taken once a week. The used transaction log extent files are backed up on at least a daily basis. On a heavily used Domino server, the used extents should be backed up several times a day. (Even on a heavily used server, the amount of data in the transactions in the log is only a small percentage of the total size of the data in the database.) All transactions against databases in Domino 6 which have transaction logging enabled are recorded in the transaction log.

  To restore a database to a point in time, you need the full backup and the transaction log extents containing transactions executed against the database since the time of the full backup. Backups are simplified since only a weekly full backup of all databases, and frequent backup of the relatively small used transaction log extent files, are needed to back up the databases, rather than full database backups on a daily basis.

- More efficient handling of disk I/O: Transaction logging may improve performance and save I/O processing time because it allows Domino to defer database updates to disk during periods of high server activity. Transactions are committed sequentially to the logfiles, typically within about 1/100 of a second of the completion of a transaction, rather than being committed to the database on disk.

  This is much quicker than database updates to random, nonsequential parts of a disk. Since writes to disk are deferred, they can be consolidated, and thus more work can be done with fewer operations. Since the transactions are already recorded, Domino can safely defer database updates until a period of low server activity.

### Transaction logging and the Domino 6 lock manager

Domino 6 replaces the semaphore method of locking a database used in Release 5 with a new lock manager. It provides for greater granularity by allowing locking at the element level, which allows for greater concurrent access and updating of the database by multiple users.

One of the most obvious effects of this is seen when databases are transactionally logged and the number of transactions increases substantially. Under R5, as a server became increasingly busy and the number of transactions increased, the response time would fall off in a direct proportion to the increase in transactions. This was due to an increase in the frequency of the flushing of changes from the universal buffer manager (UBM) to the on-disk copy of the database. During this time, the database was locked, which directly affected response time to that database.

With the implementation of object-level locking and improvements in the flushing algorithm, users are able to access the database even during the flushing operation. The net result is improved performance.

To take full advantage of the object-level locking capabilities of the new lock manager, the databases must be in Domino 6 (ODS 43) format and transaction logging must be enabled for the database. Domino 6 databases that are not transactionally logged, as well as release 5 format databases, will still use the new lock manager. However, they will not benefit from its full capabilities and will perform at a level similar to Domino release 5.

## 9.7.5 Data Protection for Lotus Domino setup

We tested a beta version of the new IBM Tivoli Storage Manager for Mail: Data Protection for Lotus Domino for Linux Version 5 Release 2 Level 1. This is the first version of Data Protection for Lotus Domino to support Linux.

In this section, we describe some of the configuration files and processes we used during our testing and present them as examples; however, this section does not contain a complete description of everything needed for installation and use of Data Protection for Domino. For that, refer to the production documentation:

http://publib.boulder.ibm.com/tividd/td/StorageManagerforMail5.2.html

Usually your Storage Administrator will install Data Protection for Lotus Domino and run the dominstall program. The dominstall program automatically configures Data Protection for Lotus Domino to work with your Domino server. This program replaces the domsetup script provided in earlier releases of Data Protection for Domino.

The dominstall program also supports multiple Domino server partitions. For more information about installing and configuring Data Protection for Lotus Domino, see *IBM Tivoli Storage Manager for Mail 5.2: Data Protection for Lotus Domino for UNIX and OS/400 Installation and User's Guide,* SC32-9056.

These are the Domino environment variables:

**DOMI_DIR** Directory where Data Protection for Lotus Domino was installed.The default installation directory is /opt/tivoli/tsm/client/domino/bin.

**DOMI_LOG** Directory where the Data Protection for Lotus Domino log file (domdsm.log) will be stored. The default is the directory where Data Protection for Domino is installed. Specify this variable to change the default setting.

**DOMI_CONFIG** File name of the Data Protection for Lotus Domino preferences file. The default is domdsm.cfg in the directory where Data Protection for Domino is installed.

Tivoli Storage Manager environment variables:

**DSMI_DIR** Directory where the Tivoli Storage Manager API is installed. This environment variable is required and there is no default.

**DSMI_LOG** Directory where the Tivoli Storage Manager API error log file (dsierror.log) will be stored. The default directory is the directory where Data Protection for Domino is installed. Use this variable to change the default setting.

**DSMI_CONFIG** The Tivoli Storage Manager API options file name. The default is the dsm.opt file.

Refer to the README file that comes with the product for the requirements and specific details for installing Data Protection for Lotus Domino on Linux.

We set up our environment for testing Data Protection for Lotus Domino for Linux on zSeries using these settings for the Domino environment variables:

```
export DOMI_DIR=/opt/tivoli/tsm/client/domino/bin
export DOMI_LOG=/domserva/notesdata
export DOMI_CONFIG=/domserva/notesdata/domdsm.cfg
```

These were our settings for the TSM environment variables:

```
export DSMI_DIR=/opt/tivoli/tsm/client/api/bin
export DSMI_LOG=/domserva/notesdata
export DSMI_CONFIG=/domserva/notesdata/dsm.opt
```

**Important:** The most common reason for problems with starting Data Protection for Lotus Domino is that the environment variables have not been set correctly. You need to ensure that all Domino and TSM variables have been defined.

Data Protection for Lotus Domino uses three files to store configuration information:

**domdsm.cfg** The Data Protection for Domino preferences file. It contains parameters specific to Data Protection for Domino.

**dsm.opt** The Tivoli Storage Manager client options file. It identifies the TSM server to contact and the node name by which Data Protection for Domino is known to the TSM server.

**dsm.sys** The Tivoli Storage Manager system options file. It contains stanzas to identify the TSM server to access. It also specifies backup and restore processing options and communication parameters.

**Note:** If you are using the TSM backup-archive client together with Data Protection for Domino, it is recommended that you define separate stanzas in the dsm.sys system options

file for the two clients. It is also recommended that you use different node names for the two clients.

The dsm.sys file that we used is shown in Example 9-19:

*Example 9-19 Sample dsm.sys file*

```
domserva@linuxa:/opt/tivoli/tsm/client/api/bin > cat
/opt/tivoli/tsm/client/domino/bin/dsm.sys
   ************************************************************************
   * Tivoli Storage Manager                                               *
   *                                                                      *
   * Sample Client System Options file for UNIX (dsm.sys.smp)             *
   ************************************************************************

      SErvername  domserva
      COMMmethod         TCPIP
      TCPPort            1500
      TCPServeraddress   wtscmxa.itso.company.com
      PASSWORDACCESS PROMPT

      NODEname            linuxa
   *   Password            linuxa

      SCHEDLOGRetention 30
      ERRORLOGRetention 30

   DOMAIN   /    /domserva/notesdata

   EXCLUDE  *
   INCLUDE  /etc/passwd
   INCLUDE  /etc/group
   INCLUDE  /domserva/notesdata/mail1/*.nsf
```

The dsm.opt file that we used is shown in Example 9-20:

*Example 9-20 Sample dsm.opt file*

```
domserva@linuxa:/domserva/notesdata > cat dsm.opt
   *======================================================================*
   * Tivoli Data Protection for Lotus Domino                              *
   ************************************************************************
   SErvername domserva
   Domain     /domserva/notesdata

    *  Quiet
      Subdir  YES
      Tapeprompt NO
```

## 9.7.6 Data Protection for Lotus Domino operations

We review some of the main command line commands for Data Protection for Lotus Domino to query, back up. and restore your Domino databases with sample results from our server.

### *Query commands*

To display the current values for Data Protection for Domino, issue **`domdsmc query preferences`**, as shown in Example 9-21 on page 258.

*Example 9-21 domdsmc query preferences*

```
domserva@linuxa:/domserva/notesdata > domdsmc q pref
Data Protection for Domino Preferences
------------------------------

BUFFers ........................ 3
BUFFERSIze ..................... 1024
DATEformat ..................... 0
LOGFile ........................ domdsm.log
LOGPRUne ....................... 60
MOUNTWait ...................... Yes
NOTESInipath ................... /domserva/notesdata
NUMberformat ................... 0
REPlace ........................ Yes
STATistics ..................... No
SUBDir ......................... No
TIMEformat ..................... 0
```

**Note:** To change a configuration parameter, you can use the **set** command. For example, to change the date format, issue: **domdsmc set DATEformat=1**.

To display general information about the Domino Server, we issued **domdsmc q domino**. A portion of the output is shown in Example 9-22.

*Example 9-22 domdsmc q domino command*

```
domserva@linuxa:/domserva/notesdata > domdsmc q domino
   IBM Tivoli Storage Manager for Mail:
   Data Protection for Lotus Domino
   Version 5, Release 2, Level 1.0
   (C) Copyright IBM Corporation 1999, 2003. All rights reserved.

   Domino Server Information
   -------------------------

     Domino Server Name:  DomServA
     Domino Server Level: 6.0
     Domino Server Build: 194
     Logging:             Archival
```

To display general information about the Tivoli Storage Manager Server, we issued **domdsmc query adsm**, as shown in Example 9-23.

*Example 9-23 domdsmc q adsm command*

```
domserva@linuxa:/domserva/notesdata > domdsmc q adsm
   /proc/sys/fs/file-max has been set to the recommended value of 131072
   /proc/sys/net/ipv4/tcp_fin_timeout has been set to the recommended value of 15
   /proc/sys/net/ipv4/tcp_max_syn_backlog has been set to the recommended value of 16384
   /proc/sys/net/ipv4/tcp_tw_reuse has been set to the recommended value of 1
   /proc/sys/net/ipv4/ip_local_port_range has been set to the recommended value of 1024
   65535
   /proc/7932/mapped_base has been set to the recommended value of 134217728

   IBM Tivoli Storage Manager for Mail:
   Data Protection for Lotus Domino
   Version 5, Release 2, Level 1.0
   (C) Copyright IBM Corporation 1999, 2003. All rights reserved.
```

```
Please enter the password for node LINUXA:
Tivoli Storage Manager Server Connection Information
---------------------------------------------------
Nodename ............................. LINUXA
NetWork Host Name of Server ........... WTSCMXA.ITSO.COMPANY.COM
TSM API Version ...................... Version 5, Release 1, Level 5
Server Name .......................... ADSM
Server Type .......................... MVS
Server Version ....................... Version 4, Release 2, Level 1.0
Compression Mode ..................... Client Determined
Domain Name .......................... DOM390
Active Policy Set .................... STANDARD
Default Management Class ............. STANDARD
```

To display a list of Domino database backups stored on the TSM server, issue **domdsmc q dbbackup**, as shown in Figure 9-9 on page 260.

```
domserva@linuxa:/domserva/notesdata > domdsmc q dbbackup

IBM Tivoli Storage Manager for Mail:
Data Protection for Lotus Domino
Version 5, Release 2, Level 1.0
(C) Copyright IBM Corporation 1999, 2003. All rights reserved.

Please enter the password for node LINUXA:


                            Database Backup List
                            --------------------

    Domino Server: DomServA
    --------------

   DB Backup Date         Size      A/I  Logged  Database Title   Database File
---------------------  -----------  ---  ------  --------------   -------------
08/29/03   13:59:47       448.00KB   A    Yes    Java AgentRunne  AgentRunner.nsf
08/29/03   13:59:48     9,984.00KB   A    Yes    Activity Trends  activity.nsf
08/29/03   13:59:51     2,110.50KB   A    Yes    Administration   admin4.nsf
08/29/03   13:59:52     1,170.00KB   A    Yes    Local free time  busytime.nsf
08/29/03   13:59:53     2,304.00KB   A    Yes    Catalog (6)      catalog.nsf
08/29/03   13:59:54       384.00KB   A    Yes    ibm's Certifica  certlog.nsf
08/29/03   13:59:55     1,300.50KB   A    Yes    Server Certific  certsrv.nsf
08/29/03   13:59:56     2,304.00KB   A    No     Domino Director  dbdirman.nsf
08/29/03   13:59:57       512.00KB   A    Yes    DCR Test         dcrtest.nsf
08/29/03   14:04:30        22.25MB   A    Yes    Mail (R6)        mail1.nsf
08/29/03   14:04:39        21.50MB   A    Yes    Mail (R6)        mail10.nsf
08/29/03   14:04:47        22.50MB   A    Yes    Mail (R6)        mail100.nsf
08/29/03   14:04:56        21.50MB   A    Yes    Mail (R6)        mail101.nsf
08/29/03   14:05:06        21.75MB   A    Yes    Mail (R6)        mail102.nsf
08/29/03   14:05:14        26.75MB   A    Yes    Mail (R6)        mail103.nsf
08/29/03   14:05:26        22.25MB   A    Yes    Mail (R6)        mail104.nsf
08/29/03   14:05:35        21.25MB   A    Yes    Mail (R6)        mail105.nsf
08/29/03   14:05:44        21.75MB   A    Yes    Mail (R6)        mail106.nsf
08/29/03   14:05:53        21.50MB   A    Yes    Mail (R6)        mail107.nsf
08/29/03   14:06:02        21.50MB   A    Yes    Mail (R6)        mail108.nsf
08/29/03   14:06:11        22.25MB   A    Yes    Mail (R6)        mail109.nsf
08/29/03   14:06:21        26.25MB   A    Yes    Mail (R6)        mail11.nsf
```

*Figure 9-9 domdsmc q dbbackup*

### *Backup commands*

To back up a specific database, enter **`domdsmc selective`**, as shown in Figure 9-10 on page 261.

```
domserva@linuxa:/domserva/notesdata > domdsmc selective "mail79.nsf"
Backing up database mail79.nsf, 1 of 1.
Full: 0   Read: 23330816  Written: 23330816  Rate: 1,313.65 Kb/Sec
Backup of mail79.nsf completed successfully.


Total Domino databases inspected:         1
Total Domino databases backed up:         1
Total Domino databases excluded:          0

Throughput rate:                          1,313.58 Kb/Sec
Total bytes transferred:                  23,330,816
Elapsed processing time:                  17.34 Secs
```

*Figure 9-10 domdsmc selective*

**Note:** If you issue **domdsmc selective** *, it will back up all databases.

To back up the transaction log files when using archival logging, use the **domdsmc archivelog** command, as shown in Figure 9-11.

```
domserva@linuxa:/domserva/notesdata > domdsmc archivelog
Starting Domino transaction log archive...
Initializing Domino connection...
Logging on to the Tivoli Storage Manager server, please wait...
Please enter the password for node LINUXA:


Archiving transaction log file /domserva/notesdata/translog/S0000054.TXN
Full: 0   Read: 67109888  Written: 67109888  Rate: 2,696.00 Kb/Sec
Archive of /domserva/notesdata/translog/S0000054.TXN completed successfully.

Archiving transaction log file /domserva/notesdata/translog/S0000055.TXN
Full: 0   Read: 67109888  Written: 67109888  Rate: 2,489.91 Kb/Sec
Archive of /domserva/notesdata/translog/S0000055.TXN completed successfully.

Archiving transaction log file /domserva/notesdata/translog/S0000056.TXN
Full: 0   Read: 67109888  Written: 67109888  Rate: 2,423.80 Kb/Sec
Archive of /domserva/notesdata/translog/S0000056.TXN completed successfully.


Total Domino transaction log files ready for archive: 3
Total Domino transaction log files archived:          3

Throughput rate:                                      2,531.40 Kb/Sec
Total bytes transferred:                              201,329,664
Elapsed processing time:                              77.67 Secs
```

*Figure 9-11 domdsmc archivelog*

You can also back up the transaction log files using a threshold. In Example 9-24, we used a 70% threshold.

*Example 9-24 domdsmc archivelog with threshold*

```
domserva@linuxa:/domserva/notesdata > domdsmc archivelog -adsmpwd=mypassword
-threshold=70,0
Starting Domino transaction log archive...
Initializing Domino connection...
Logging on to the Tivoli Storage Manager server, please wait...


Total Domino transaction log files ready for archive: 3
Total Domino transaction log files archived:          0

Throughput rate:                                      0.00 Kb/Sec
Total bytes transferred:                              0
Elapsed processing time:                              0.00 Secs
```

### Restore commands

You can use the **domdsmc restore** command to restore database backups and optionally activate the databases, as shown in Figure 9-12 and Figure 9-13 on page 263.

```
domserva@linuxa:/domserva/notesdata > domdsmc restore "mail100.nsf" /mountwait=yes
/subdir=yes /adsmpwd=mypassword /pick=SHOWALL /into=" mail100.bak"

Starting Domino database restore...

Initializing Domino connection...
Logging on to the Tivoli Storage Manager server, please wait...
Querying Tivoli Storage Manager server for a list of database backups, please wait...

IBM Tivoli Storage Manager for Mail Scrollable PICK Window - Restore

     #    DB Backup Date              Size   A/I  Logged  Database File

-----------------------------------------------------------------------------------------
-----------------------------------
     1. | 08/29/03   14:04:47        22.50MB  A   Yes     mail100.nsf
        |
        |
        |
       |
        |
        |

0---------10--------20--------30--------40--------50--------60--------70--------80------
--90--------100------110------120--
<U>=Up  <D>=Down  <T>=Top  <B>=Bottom  <R#>=Right  <L#>=Left
<G#>=Goto Line #  <#>=Toggle Entry  <+>=Select All  <->=Deselect All
<#:#+>=Select A Range <#:#->=Deselect A Range  <O>=Ok  <C>=Cancel
pick> 1
```

*Figure 9-12 domdsmc restore*

We selected the file to restore by **pick> 1** as shown in Figure 9-12, and then typed **o** for ok to start the restore, as shown in Figure 9-13 on page 263.

```
IBM Tivoli Storage Manager for Mail Scrollable PICK Window - Restore

     #    DB Backup Date              Size   A/I  Logged  Database File

-----------------------------------------------------------------------------------------
-----------------------------------
x    1. | 08/29/03   14:04:47        22.50MB  A   Yes     mail100.nsf
        |
        |
        |
        |
        |
        |

0---------10--------20--------30--------40--------50--------60--------70--------80------
--90--------100------110------120--
<U>=Up  <D>=Down  <T>=Top  <B>=Bottom  <R#>=Right  <L#>=Left
<G#>=Goto Line #  <#>=Toggle Entry  <+>=Select All  <->=Deselect All
<#:#+>=Select A Range <#:#->=Deselect A Range  <O>=Ok  <C>=Cancel
pick> o

Restoring database mail100.nsf, 1 of 1,
  to /domserva/notesdata/mail100.bak.dad
Full: 0   Read: 23592960  Written: 23592960  Rate: 2,736.99 Kb/Sec
Restore of mail100.nsf completed successfully.


Total database backups inspected:              1
Total database backups requested for restore:   1
Total database backups restored:               1
Total database activated:                      0

Throughput rate:                               2,736.99 Kb/Sec
Total bytes transferred:                       23,592,960
Elapsed processing time:                       8.42 Secs
```

*Figure 9-13 domdsmc restore (continued)*

To bring the restored database online and apply the logs, use the **domdsmc activatedbs** command as shown in Figure 9-14 on page 264 and Figure 9-15 on page 265.

```
domserva@linuxa:/domserva/notesdata > domdsmc activatedbs /mountwait=yes /pick
/adsmpwd=mypassword /applylogs="09/02/2003,16:30:00"

Starting Domino database activation...

Initializing Domino connection...
Logging on to the Tivoli Storage Manager server, please wait...
IBM Tivoli Storage Manager for Mail Scrollable PICK Window - Activate
     Apply Logs to Time - 09/02/2003 16:30:00

     #    Backup Time Stamp           Size   A/I  Logged  Database File

------------------------------------------------------------------------------------------
-----------------------------------
     1. | 08/29/2003 13:59:25        22.50MB  A   Yes     mail100.bak
        |
        |
       |
        |
        |

0---------10--------20--------30--------40--------50--------60--------70--------80------
--90--------100------110------120--
<U>=Up  <D>=Down  <T>=Top  <B>=Bottom  <R#>=Right  <L#>=Left
<G#>=Goto Line #  <#>=Toggle Entry  <+>=Select All  <->=Deselect All
<#:#+>=Select A Range <#:#->=Deselect A Range  <O>=Ok  <C>=Cancel
pick> 1
```

*Figure 9-14 domdsmc activatedbs*

```
IBM Tivoli Storage Manager for Mail Scrollable PICK Window - Activate
     Apply Logs to Time - 09/02/2003 16:30:00

     #    Backup Time Stamp           Size   A/I  Logged  Database File

-------------------------------------------------------------------------------------
-----------------------------------
x    1. | 08/29/2003 13:59:25        22.50MB  A   Yes     mail100.bak
        |
        |
        |
        |
  |
        |
        |

0---------10--------20--------30--------40--------50--------60--------70--------80------
--90--------100------110------120--
<U>=Up  <D>=Down  <T>=Top  <B>=Bottom  <R#>=Right  <L#>=Left
<G#>=Goto Line #  <#>=Toggle Entry  <+>=Select All  <->=Deselect All
<#:#+>=Select A Range <#:#->=Deselect A Range  <O>=Ok  <C>=Cancel
pick> o

Starting transaction log file processing...

Media Recovery Replay (139 MB): 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%
09/02/2003 16:29:27   Recovery Manager: Media Recovery complete for
/domserva/notesdata/mail100.bak.dad, last update applied 09/02/2
003 02:04:14.

Transaction log file processing completed successfully.

Activating database mail100.bak, 1 of 1,
Activate of mail100.bak completed successfully.


Total pending databases inspected:                1
Total pending databases requested for activation:  1
Total pending databases activated:                 1

Total archived log backups requested for restore:  0
Throughput rate to restore log files:              0.00 Kb/Sec
Total bytes transferred to restore log files:      0
Elapsed processing time to restore log files:      0.00 Secs
```

*Figure 9-15 domdsmc activatedbs (continued)*

We recommend that you make at least one full backup (**domdsmc selective**) each week. A full backup may be required more often on a server with a large number of transactions in order to reduce the number of transaction log extents which much be processed during a restore. You must find a balance between frequent full backups and long restore time.

An archivelog with no threshold (**domdsmc archivelog**) should be done once a day, preferably off-shift, as well as a daily incremental backup (**domdsmc incremental**), which would perform full backups of nsf files for which DBIIDs have changed (due to fixup or OS operation).

You should also inactivate backed up files that do not have any transactional logs associated with them (**domdsmc inactive**). Example 9-25 shows a sample schedule for Data Protection for Lotus Domino using **crontab**.

*Example 9-25 Sample schedule for Data Protection for Lotus Domino using crontab*

```
crontab -l
###TDP Daily Archive
00 04 * * * domdsmc archivelog
###TDP Inactive Files
00 11 1 * * domdsmc inactive
###TDP Archive
01 * * * * domdsmc archivelog -threshold 70,0
###TDP Incremental
00 01 * * * domdsmc incremental
###TDP Full Backup
00 1 * * 0,2,4 domdsmc selective "*"
```

## 9.7.7 Anti-virus

An anti-virus solution was not available for testing at the time of writing. However, we could see a solution available from Trend Micro in the future. See the Trend Micro Web site for more information.

http://www.trendmicro.com/en/products/email/smln/evaluate/overview.htm

## 9.7.8 Anti-spam

An IBM Redbook devoted to the topic of avoiding spam in a Domino 6 environment is available; refer to *Lotus Domino 6 spam Survival Guide for IBM*, SG24-6930.

10

# Troubleshooting

In this chapter, we describe problems encountered while installing and using Domino and Linux, and also list their causes and solutions.

## 10.1 Domino stalled under VM

**Problem:** The Domino server stopped responding to clients and stopped writing messages to the console.

**Cause:** We were running Domino on Linux in a z/VM virtual machine. We had logged on to the virtual machine console and allowed the screen to go into HOLDING mode. This put the virtual machine into a stopped state.

**Solution:** We noticed the problem quickly, pressed Clear (ESC) on the VM console, and the virtual machine started running again. It was only stopped for 6 minutes. Domino recovered and continued processing requests. We normally do not recommend logging onto the virtual machine console while Domino is running.

## 10.2 Idle server still uses 2 to 3 percent of system

An idle server still requires some amount of storage and processor resource, as it constantly polls for new connections. The cost of this seems to be about 2 to 3% of a z800 processor. This may be a characteristic of Domino, as discussed in “Idle servers” on page 300. However, it also may be due to the Linux timer, which pops 100 times per second.

There is a patch for the Linux kernel 2.4.7 that changes the timing method from the regular 100 hz interrupts to on-demand timer interrupts. This saves some CPU cycles in an idle Linux image. The drawback is that a busy Linux image will do a small amount of additional work every time the switch between user mode and kernel mode is done. In short, if you are running many images under VM, the on-demand timers are useful but you probably do not want to use them running under an LPAR. The patch can be found at:

```
http://www-124.ibm.com/developerworks/oss/linux390/index.shtml
```

The patch introduces a configuration option CONFIG_NO_HZ_TIMER that lets you choose between the old and the new behavior. Note that you will need different lcs, qdio, qeth, and z90crypt modules if you use the no hz timer option.

In earlier versions of the timer patch, kernels that were built with the patch applied but with CONFIG_NO_HZ_TIMER switched off were still incompatible with the regular OCO modules (and also with the timer-patch OCO modules).

This problem is fixed in this version of the patch. If CONFIG_NO_HZ_TIMER is switched off, the kernel is now compatible with the regular OCO modules.

**Note:** The “order 2 allocation relief” patch for 64-bit is not compatible with the “on-demand timer” patch.

The value of the timer patch status is in /proc/sys/kernel/hz_timer. These are the values:

```
1 - run free at 100hz
0 - sets the timer for on-demand switching for running under VM
```

To enable the timer patch, issue the following command and put it in /etc/rc.d/boot.local:

```
echo "0" >/proc/sys/kernel/hz_timer
```

This patch is also discussed in “The Linux timer patch” of the redbook *Linux on IBM @server zSeries and S/390: Performance Measurement and Tuning,* SG24-6926.

## 10.3 Domino Console message: Insufficient TCP sockets

**Problem:** During a test run with 1500 simulated Notes users, the system started processing the first users, but we then got multiple messages on the Domino console saying: `Insufficient TCP sockets`. Many of the Notes clients showed messages saying they were unable to connect to the server.

**Cause:** TCP sockets on Linux relate to file descriptors. When we issued **`ulimit -a`** from the user ID that started Domino, we saw the number of open files (-n) was 1024. This should have been set higher, as in the installation instructions. We think this problem occurred because we logged in as the root user, then switched to domserva using **`su domserva`**. This does not run the ssh daemon, and so it did not read the limits.conf file.

**Solution:** We changed limits/conf so the number of file descriptors applies to all users, not just domserva. We confirmed this was ok by logging in as the root user, then doing **`su domserva`** and running **`ulimit -a`**. It showed the expected number of open files: 20000 soft, and 49152 hard.

For information about a similar situation, refer to “File descriptors” on page 136.

## 10.4 Translog placed in wrong directory

**Problem:** We discovered that the translog was placed in the wrong directory.

**Cause:** The default value of /translog caused it to be placed in the root, because of the leading slash.

**Solution:** Any directory that is to be placed inside another one must be specified *without* the slash (/), for example: translog, not /translog. Or the full path must be specified, for example: /notesdata/translog.

## 10.5 DB2: Starting the database manager

**Problem:** We received the following error messages (not all at once).

```
Server Error:  Realtime Service cannot connect to external system
SQL1013N  The database alias name or database name "DB2DB" could not be found.
SQLSTATE=42705
SQL1032N  No start database manager command was issued.  SQLSTATE=57019
```

**Cause**: DB2 might not be started.

**Solution:** Here is how to start the DB2 database manager. It assumes that you have a sample DB2 database named “sample”.

1. Login onto your Linux guest with your DB2 user ID, such as db2inst1. If you are already logged in, issue the switch user command: **`su - db2inst1`**
2. Issue the command: **`db2start`**
3. Issue the command **`start dbm`** to start the database manager.
4. To see the database named sample, issue: **`connect to sample`**
5. To see if all the DB2 processes are running, type: **`ps-ef |grep DB2`**

## 10.6 DECS: Starting and testing connectivity

**Problem:** You cannot get your DECS or DCR session to connect to a database.

**Cause:** DECS must be running in order to communicate to DB2 with it, or with DCRs.

**Solution:**

- You can include DECS in the list of started Domino processes in notes.ini, as follows:

  `ServerTasks=.....,DECS,...`

- If you want to start DECS manually from the Domino console, issue: **`load decs`**

  To test for connectivity to DB2, complete the following steps:

  a. Run the test program DCTEST, located in the Domino program directory. On our platform, the program directory was located in /opt/lotus/notes/latest/zlinux, the default directory. This test should be conducted using the Domino user/account ID.

  b. Select **DB2** from the program menu.

  c. Enter the data source, user name, and password when the program prompts for them.

  d. Choose (Y/N) whether you want detailed driver information.

  e. You can produce a printed report for diagnostic purposes. When asked, you can choose to output to a file (Y) or not (N). If you do not choose output to a file, the results appear on your monitor.

  f. If you chose file output, supply a name for the file, then press Enter. The program then attempts to connect to the DB2 data source.

  g. You can retry a connection by entering: `Y` when prompted for another data source.

## 10.7 The DECS hang

**Problem:** We were able to seemingly crash the Domino server (actually, it was locked up) by accessing LotusScript functions that did look-ups external to the server itself, but with the server's address incorrectly specified in the server doc in the NAB.

**Cause:** The server setup routine placed the vmlinuxa into the server doc by default. vmlinuxa was not in the DNS, and either the LotusScript interpreter or some API call could not resolve that name to the numeric IP address. This caused the server to lock up.

**Solution:** When we changed the entry in the server doc from vmlinuxa.itso.company.com to the numeric IP address, everything functioned properly.

## 10.8 A Domino panic

We thought you might be interested in viewing a Domino panic listing.

```
Lotus Domino (r) Server, Build V65_07292003, July 29, 2003
Copyright (c) IBM Corporation 1987, 2003. All Rights Reserved.

PANIC_SQLO _rc     = 0x00000000
PANIC_SQLO _hiCode = 0x00000102
PANIC_SQLO errno?  = 0x0000000D
PANIC_SQLO file    = logger/amqhsofn.c
Unexpected OS error returned to logger:
2C:6Er error: comp = 11, fnc = 81, probeid = 79, errcode = 5010, extsympt =
```

```
Unexpected internal error returned to logger: 0xA0772010
Logger Failure: Func=hlgStartLog RC=0x00002010 File=rm/rmrest.cpp:828
Recovery Manager: Unable to recover DB /domserva/notesdata/statrep.nsf
Recovery Manager: Unable to recover DB /domserva/notesdata/names.nsf
Recovery Manager: Unable to recover DB /domserva/notesdata/log.nsf
Recovery Manager: Unable to recover DB /domserva/notesdata/events4.nsf
Recovery Manager: Unable to recover DB /domserva/notesdata/clubusy.nsf
Recovery Manager: Unable to recover DB /domserva/notesdata/cldbdir.nsf
Recovery Manager: Unable to recover DB /domserva/notesdata/admin4.nsf
Unable to complete restart processing, err = Recovery Manager: Error from Logging
Subsystem..

 Thread=[00757:00002-01024]
Stack base=0x7FFEE6C8, Stack size = 2696 bytes
PANIC: Unable to complete restart processing.
0822C068  unknown! unnamed +0
0822BAD6 /opt/lotus/notes/latest/zlinux/libnotes.so! OSWalkStack +126
08228D42 /opt/lotus/notes/latest/zlinux/libnotes.so! Panic +366
0882CDA4  unknown! unnamed +0
08A9217E  unknown! unnamed +0
088285CC  unknown! unnamed +0
0881D8A6 /opt/lotus/notes/latest/zlinux/libnotes.so! NSFDbSetMode +130
08ACB1DE /opt/lotus/notes/latest/zlinux/libnotes.so! NIFNSFInitialize +66
0046D988 /opt/lotus/notes/latest/zlinux/server! MainServerInit +1292
004211DC /opt/lotus/notes/latest/zlinux/server! FirstProcessMain +40
00421176 /opt/lotus/notes/latest/zlinux/server! ServerMain +890
0042OB76 /opt/lotus/notes/latest/zlinux/server! main +190
099B27BC /lib/libc.so.6! __libc_start_main +176
0042O9DC /opt/lotus/notes/latest/zlinux/server! OSUpdateFileStatistics +88
Stack base = 0x7ffee6c8, Stack size = 4064 bytes
Fatal Error signal = 0x0000000b PID/TID = 757/1024 LWP = 757
FAILING PSW = 070dc000 899c661e  /lib/libc.so.6(kill+0x2) [0x99c661e]
 R0=00000001  R1=099c661c  R2=00000000  R3=0000000b
 R4=8983c3dc  R5=098410cc  R6=7ffee6c8  R7=00000000
 R8=000013e3  R9=0000000b R10=000002f5 R11=00000400
R12=09847b7c R13=8983c348 R14=8983c3e8 R15=7ffedbe8
0822C068  unknown! unnamed +0
0822BAD6 /opt/lotus/notes/latest/zlinux/libnotes.so! OSWalkStack +126
081E943A  unknown! unnamed +0
0983C4EA  unknown! unnamed +0
7FFEDA30  unknown! unnamed +0
0983C3DC /lib/libpthread.so.0! pthread_kill +156
08228E14 /opt/lotus/notes/latest/zlinux/libnotes.so! Panic +576
0882CDA4  unknown! unnamed +0
08A9217E  unknown! unnamed +0
088285CC  unknown! unnamed +0
0881D8A6 /opt/lotus/notes/latest/zlinux/libnotes.so! NSFDbSetMode +130
08ACB1DE /opt/lotus/notes/latest/zlinux/libnotes.so! NIFNSFInitialize +66
0046D988 /opt/lotus/notes/latest/zlinux/server! MainServerInit +1292
004211DC /opt/lotus/notes/latest/zlinux/server! FirstProcessMain +40
00421176 /opt/lotus/notes/latest/zlinux/server! ServerMain +890
0042OB76 /opt/lotus/notes/latest/zlinux/server! main +190
099B27BC /lib/libc.so.6! __libc_start_main +176
0042O9DC /opt/lotus/notes/latest/zlinux/server! OSUpdateFileStatistics +88
Tue Aug 19 10:01:27   Running NSD
NSD is in progress .................

Please attach the following files to your bug report along with the server log:
Log file                :
/domserva/notesdata/IBM_TECHNICAL_SUPPORT/nsd_all_Linux_linuxa_08_19@10_01.log
```

```
Domino Halted. Type 'quit' to Terminate/Cleanup quit

Tue Aug 19 10:02:53   Termination is in progress
Tue Aug 19 10:02:53   Terminating tasks
Tue Aug 19 10:02:58   Freeing resources
Tue Aug 19 10:02:58   Termination completed
domserva@linuxa:/domserva/notesdata>
```

11

# Capacity planning for Linux on zSeries

*Capacity planning* is the science of taking existing knowledge about resource requirements, and projecting future needs based on known changes to workload or environment. When companies plan a consolidation to move many servers to Linux on zSeries, that infrastructure will have a great impact on resource requirements.

In this chapter, we discuss some of the choices in infrastructure, and the impact on capacity requirements. We describe capacity planning guidelines in the areas of:

- Processor
- Storage
- Domino partitioned servers (DPARs)
- zSeries logical partitions (LPARs)
- zSeries z/VM guests
- Network
- Disk/DASD
- Special Domino considerations

## 11.1 Recommendations

These are our general recommendations for capacity planning:

- Estimate carefully.

  Initial estimates of the resources required to support a Domino server are based on estimates of various factors, including:

  - Number of registered users
  - Number of connected users
  - Number of active users
  - Message rates generated by users on mail servers
  - Transaction rates generated by active users
  - Average CPU time used per active user
  - Amount of data, including the average mail file size and size of application databases

  The more accurately you estimate these values, the more accurate will be your estimate of required resources.

- Make use of the capacity planning assistance available from IBM.

  See 11.2.6, “IBM initial sizing assistance” on page 277 for instructions on requesting an initial capacity sizing.

- Base your capacity planning on your own measurements as soon as possible.

  Every Domino environment is different. Therefore, you should monitor your system as soon as you have a significant Domino workload, and then revise your plans based on that workload. If you have other Domino servers, you can collect information about your workload profile from them, and use that to help validate your assumptions.

- Remember that some guidelines presented in this book may change.

  The most recent capacity tests are based on Domino 6.5.1 running on z900 and z990 processors. Later Domino releases and processor models may impact certain guidelines.

- Beware of running your own performance tests and misinterpreting the results.

  It is possible to set up some seemingly simple performance tests. However, we have seen very misleading results when trying to use the run time of a single Domino function to estimate the performance of a production workload. Unless you understand exactly which system resources the function uses, and can accurately relate that to the resources your production workload will use, you can get erroneous results. In addition, unless you understand how Domino performs these functions, it is possible that you are measuring something quite different from what you think. You could, for example, be measuring network performance instead of server performance. Realistic tests are, unfortunately, complex and expensive to run. Use the information in this document to get capacity estimates for production workloads.

## 11.2 Estimating processor capacity

The processor capacity required for Domino depends on:

- The number of registered users
- The number of active users
- The type of users (Notes, DWA (iNotes), POP3, IMAP, DAMO (Outlook) clients)

- The workload profiles of users (light, medium, heavy or power)
- The number of transactions generated by users
- For mail servers, the number and size of mail messages sent and delivered
- The CPU required for background tasks and non-user-driven functions
- The configuration of DPARs and LPARs/z/VM guests
- The number of HTTP hits (reads, writes) for browser users
- The size of databases and the number of entries in the views used by databases
- In addition, many other factors affect capacity and performance. These include:
  - Amount of full-text indexing of databases
  - Frequency of database searches against indexes
  - Extent of Domino agent activity
  - Replication and local replicas
  - Domino clustering
  - Type-ahead name searches
  - Other special functions and features of Domino (encryption, network compression, Message Tracking, etc.)
  - Implementation of third-party software

### 11.2.1 How users impact processor capacity

The number of registered users, how active the users are, and the CPU time used per active user can be used as a basis for estimating part of the required capacity. The calculation is the number of active users times the CPU used per active user during some time interval. For example, for a one hour interval, the percent utilization of the processor is: (CPU seconds used by active users per hour/number of CPs/3600) * 100%.

Other components of the CPU consumed by a DPAR are not necessarily tied to active users in the DPAR. For example, on mail servers, users receive mail whether or not they have accessed the server. For mail servers, we recommend tracking the number of messages generated by registered users either on a daily or prime shift basis. Additionally, Domino servers run background tasks, such as replication and administration processes, which are not dependent on the number of active users. For a low number of active users, these background tasks account for most of the CPU used by the DPAR. It is important to understand the low utilization effect. If you take the CPU used by a server with 50 users, then 500 users will not use 10 times the CPU because the cost of background tasks would not go up ten-fold.

If you do not currently run Domino on zSeries, begin by contacting your IBM marketing representative to get an initial capacity estimate. See 11.2.6, “IBM initial sizing assistance” on page 277 for instructions on requesting an initial sizing.

### 11.2.2 Which user counts are important?

There are a number of user counts for Notes Remote Procedure Call (NRPC) clients supplied by the Domino server, among them: connected users, instantaneous active users, and active (1, 3, 5, 15, 30) minute users. We use the active 15 minute users count as a measure of the activity on a Domino server. As you will see, this metric more accurately influences the CPU consumed by the Domino server than many other metrics. Note that this Domino statistic does not apply to iNotes clients who use the HTTP rather than NRPC protocol. The activity of

iNotes clients is based on the number of HTTP hits, which is available as Domino HTTP read and write statistics.

On other platforms, the number of connected or even registered users may be used as the measurement of activity. These counts are typically significantly higher than the active 15 minute user count. When making cross-platform comparisons, it is important to ensure that like metrics are used.

#### Registered users

You can get the number of registered users on your server relatively easily for mail applications; it is the number of users whose mail files are located on a server. For other Domino application servers, it is the total number of unique users authorized for databases on a server.

The number of registered users is not reported in any Domino statistic. However, it is usually known by the Domino administrator.

The number of registered users and the average mail database size per user can be used to estimate the disk space required for user mail files.

#### Connected users

Connected users have been active recently, and have not been inactive long enough to have timed out on the server. The connected count is available from the Domino statistic Server.Users. This statistic only applies to NRPC clients; browser mail clients use connection-less protocols. The number of connected users is influenced by the user time-out parameter Server_Session_Timeout in notes.ini. For example, changing this value to 60 minutes from the default of 4 hours could easily drop the number of connected users by 50%, and also decrease storage requirements because connected users consume some memory. A suggested Server_Session_Timeout value is between 30 and 60 minutes.

#### Active users

Active users are those users who have been active in the previous specified period. For NRPC clients, we recommend that you use the number of active users in the previous 15 minutes, which is available from the Domino statistic Server.Users.Active15Min. The active user rate is important because it influences the CPU and processor storage required for the server. Active 15 minute users divided by the number of registered users provides an indicator of how busy a server is.

### 11.2.3 Client types

Identifying the client types for active users is very important in projecting capacity. Some clients consume considerably more CPU than others. For example, DWA (iNotes) and Webmail clients use between three and four times as much CPU as Notes clients because most of the workload has been transferred to the Domino server. This is in no way a zSeries phenomenon, but a cross-platform statement. Other clients, such as Domino Access for Microsoft® Outlook (DAMO), POP3, and IMAP are much closer in cost to Notes clients. When projecting capacity, it is important to accurately quantify how many of each client type will be accessing a Domino server during peak processing periods.

### 11.2.4 Domino transaction rates

A Domino transaction is not a user interaction. It is a Domino call to a high level function, such as opening a database. Thus, one user interaction will involve several Domino transactions. Only clients using the NRPC protocol generate transactions; browser clients do not generate

transactions. The rate at which transactions are processed by the Domino server depends on the number of active NRPC users and the rate at which they are working. For example, if an installation with 1000 registered users has an average of 200 active users during the hour, each working at a rate of 90 server transactions per hour, then the server would need to process 18,000 transactions per hour.

Note that the transaction rate per user could be higher during a pilot when users are experimenting with Domino functionality. Alternatively, the transaction rate per user could be lower until a critical mass of users is achieved on the server.

The number of transactions divided by the number of active 15 minute users gives a measure of how busy users are when accessing the server. Transaction rates are a valid metric to track Domino server activity as long as there are no changes to server/client releases. The number and kinds of transactions will vary with Domino server and client releases. For example, although most customers saw CPU reductions when moving from Domino R5 to 6/6.5, Domino 6/6.5 transaction rates were observed to be higher than R5 transaction rates.

The Domino statistic for transaction count is Server.Trans.Total, which is the total number of transactions since server initialization. Before computing a transaction rate, computations must be performed to determine the correct number of transactions for the desired interval.

### 11.2.5 Consider the peak times

There are times during the day when the capacity needed by Domino is higher than the average, and there are times when it is lower. If Domino will represent less than 50% of the total workload on the zSeries server, and where the other workloads are of lower priority, you may be able to use the average workload for planning. Domino peaks may not coincide with peaks for other workloads, in which case it is important to consider the consolidated peak for all of the workloads when planning capacity. For dedicated Domino servers or on systems where Domino is not the highest priority, a peak load factor should be considered to ensure consistently good response times.

Determining the peak load factor to use requires some understanding of the work schedule of the users. If the users are teachers who can only use the system between classes or at the beginning/end of the day, then the peak load factor could be much higher than two times the average. If the majority of the users are using a local copy of their mail database on their workstation, then the peak load factor could be much less than two times the average. Also, if users span multiple time zones, then there will be lower peaks but during a wider window. Seasonal and business trends may also influence the measurement number. For example, there may be an increase of 30% in active users and transactions from August to September as people return from summer vacations. The use of a local replica by a high percentage of users results in a more uniform workload throughout the business day. But users may start their replication earlier during the workday.

If you are migrating Domino users from other platforms, you may be able to determine the peak times and peak-to-average ratios by analyzing Domino and platform statistics.

### 11.2.6 IBM initial sizing assistance

An initial capacity planning estimate for a Domino workload on Linux on zSeries is available through your IBM marketing representative. He/she can request a sizing questionnaire from IBM TechXPress, complete the questionnaire with your help, and submit it. TechXPress will route it to the organization which does the sizings for Domino on zSeries.

The following types of information are needed to complete this form:

- Number of registered users
- Number of connected users
- Peak number of active users in a 15 minute period by usage pattern (light, medium, heavy, power) and client type (NRPC (Notes), DWA (iNotes), POP3, IMAP, Webmail, DAMO (Outlook)
  Definitions are provided for the user profiles of light, medium, heavy and power.
- Exploited functions/features of Domino: clustering, full-text indexing, personal agents, Message Tracking, local replicas, port encryption, network compression, SSL, and so on
- Projection of future mail growth
- Description of current production environment
- Requirements for third-party tools, like anti-virus

Upon receipt of the completed questionnaire, the sizer will perform the capacity sizing and then provide the marketing representative with an estimate of the processor size you will need to support this workload. IBM employees can submit a TechXpress request on the IBM intranet or by phone. Find out more at this Web site:

http://w3-1.ibm.com/support/techxpress.html

## 11.3 Capacity and performance testing for this redbook

Domino 6.5 is the first release of Domino to run on Linux on zSeries, either natively, or in an LPAR, or under z/VM. In this chapter, we describe the process for performing capacity planning for Domino on Linux workloads based on knowledge of the environment, some initial tests performed during the redbook project, and some larger and more in-depth tests performed at a later time by the IBM Linux Scalability Center (LSC).

### 11.3.1 Initial testing

All initial testing for the redbook was done with Domino mail servers running in z/VM guests on a z800 with two CPs, using virtual disk for swapping. SuSE SLES8 SP2 and z/VM 4.3. were the software releases used for the testing.

For experimentation purposes, three Linux servers were used with different configurations to show specific points of interest:

- LinuxA supported one Domino server, and was used for experiments running between 300 and 1500 users. Users were simulated with Notesbench mail clients. The DASD for LinuxA was ESS on FICON® channels.
- LinuxB supported one Domino server, and ran 400 users. The DASD for LinuxB was RVA on ESCON® channels.
- LinuxC supported two Domino servers, each with 400 users. DASD for LinuxC was mixed.

The purpose of the different environments was to test memory constraints, understand DASD performance issues, and determine any side effects of running servers together. All of the tests were performed using two virtual processors backed by two real processors. Thus, each Linux server had two virtual processors defined.

We ran one workload with just one virtual processor; it had very poor performance. We did not research this since it is a cross-platform recommendation to run Domino with two or more processors. Access to two processors avoids the potential of elongated response times during periods when multiple Domino processes are contending for a single CP. For performance reasons, it is also recommended to never define more virtual CPs than available real CPs for a Linux guest.

Many of the results of this initial testing are documented in 11.9, “Capacity monitoring” on page 294, which provides examples of capacity monitoring as an ongoing activity to project capacity requirements.

### 11.3.2 LSC testing

The LSC testing was performed over a period of several months and was completed almost one year after the initial redbook testing. Capacity planning recommendations in this redbook have been updated with the results of this latest testing. These recommendations are valid for the current 6.5.x releases of Domino, but may change with Domino 7 and later releases.

The objectives for the LSC testing were to:

- Provide data to assist in sizing resource requirements for Domino mail servers deployed on zSeries Linux running in LPARs and z/VM guests
- Identify configuration and tuning recommendations to optimize throughput of Domino mail implementations on zSeries Linux.

The following server, storage, and disk setup was used:

- zSeries z900 Type 2064 Model 116
  - 2 LPARs running Linux with 8 CPs each, configureable as one 16-way as needed There were also some runs with more LPARs and fewer CPs.
  - z/VM with guests running Linux 64 GB total
  - Guest storage varied with workload
  - z/VM guests running Linux with two or more logical CPs per guest

    Testing was done with combinations of 1 to 16 guests.
- 2 DASD 2105 (ESS Model 800)
  - 5 TB

For the network, an OSA Express Gigabit LAN was implemented. For the software, the LSC installed SuSE SLES 8 SP3 and z/VM 4.4 with APAR VM63385, which alleviates duplicate SIGA problems. These releases are the currently recommended minimum software releases for Domino for Linux on zSeries.

Notes and iNotes users were simulated with Lotus Notesbench R6mail clients. The LSC testing showed that Domino servers running in Linux LPARs and z/VM guests can support several thousand Notesbench users if properly configured. However, keep in mind that Notesbench users are not equivalent to production users. The processing scenarios for Notesbench clients do not include all the functions/features that production users might access. Additionally, Domino servers used for Notesbench testing are frequently brought up with a minimal configuration of 2 to 3 Domino tasks. Most Domino production servers will be running at least 10 to 20 tasks. Lastly, Notesbench users are constantly accessing the server with a steady amount of work; this is not comparable to the ebb and flow of a production workload.

Later sections of this chapter include recommendations on optimal configurations to efficiently exploit processor and storage capacity.

## 11.4 Processor planning

One of the key components in determining processor requirements is the number of active users at peak processing periods. Another important component of the equation is the type of client. The LSC ran a number of tests, varying the number of active Notes and iNotes clients, with Domino running in different LPAR and z/VM guest configurations.

### 11.4.1 How many CPs is too many?

The LSC executed several scenarios to test the scalability of Domino running in Linux LPARs. During the runs, the number of Notes and iNotes users were pushed to a maximum to determine how many users can be supported by client type in a Domino partition (DPAR) and LPAR. The number of CPs available to the LPARs was varied from 1 to 8 CPs. The results showed that Notesbench iNotes clients scale very well up to 8 CPs. On the other hand, Notesbench Notes clients do not scale as well; issues became apparent somewhere between 2 and 4 engines. The LSC's analysis confirmed that the differences in scalability can be directly attributed to Linux swapping rates (pswpins) as illustrated in Figures Figure 11-1 on page 281 and Figure 11-2 on page 281.

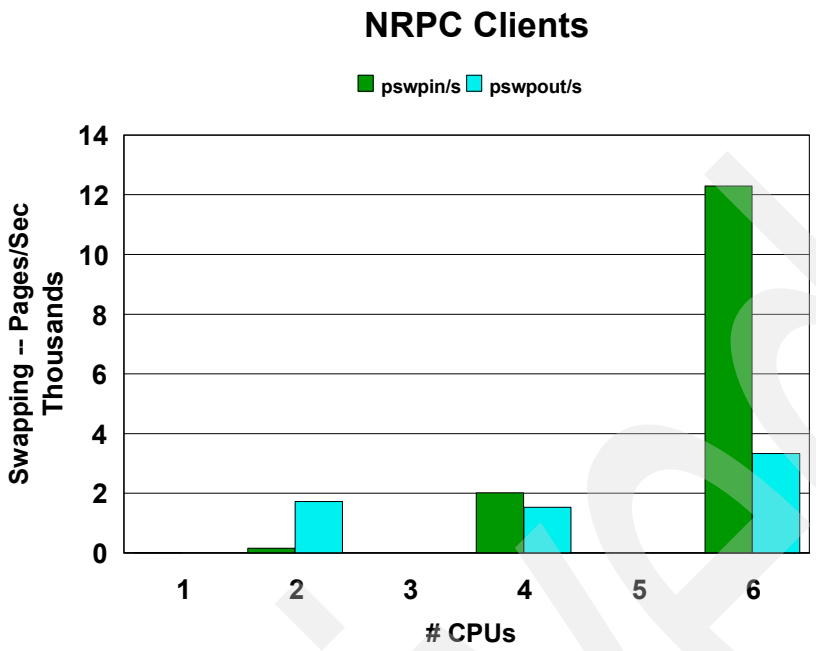



*Figure 11-1 NRPC clients*

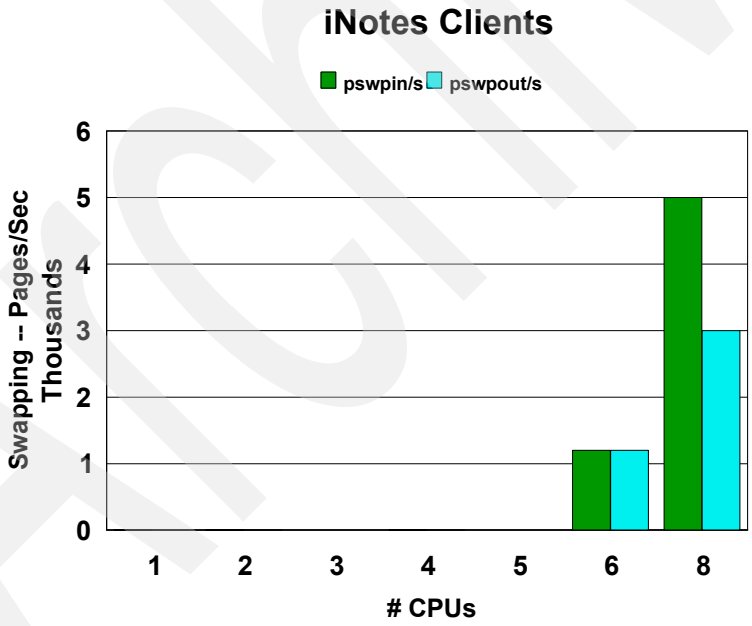



*Figure 11-2 iNotes clients*

The conclusions were that a more CPU-intensive workload such as iNotes consumes storage at a slower rate than Notes, and therefore is able to exploit available CPU more efficiently. Notesbench Notes users are extremely light users of CPU. When a Domino server(s) is driven with very high numbers of lighter clients, then storage is consumed at a much faster rate than with a more CPU-intensive workload. As the number of CPs increases, there are more requests for the kernel per user. Although the processors are busy, they are swapping and not performing useful work in terms of Domino throughput.

The Notes scalability test illustrates a good point about Domino's intolerance for paging/swapping, but its results must be viewed in a proper context. The Notesbench NRPC workload is much lighter than most production Notes workloads. Many production Notes workloads fall somewhere between the Notesbench Notes and iNotes workloads; some production workloads are more CPU intensive than the Notesbench iNotes workload. When projecting capacity, it is very important to understand the work habits of the targeted audience. Are the users very light users or medium to heavy users? Although less CPU-intensive users will require fewer engines, they also will change the dynamics of how those engines are exploited in a storage constrained system.

There are no issues with Domino NRPC or for that matter any other client's scalability if Domino is configured to minimize Linux swapping. Domino 6.5.x releases currently only run in a 31-bit Linux kernel, which means that each Linux image is limited to 2 GB of central memory and some amount of expanded memory/virtual disk if running under z/VM. This limitation will go away with Domino 7 when 64-bit as well as 31-bit Linux are supported environments. Domino itself is a 31-bit application, and there are no plans to convert it to 64-bit addressing.

The LSC performed a number of additional measurements for the NRPC workload running in Linux LPARs. The LPARs were limited to 2 CPs and the maximum number of users that could be supported without Linux kernel swapping. The scalability results were dramatically improved. Figures Figure 11-3 and Figure 11-4 on page 283 show a before and after picture where Domino is compared to standard Large System Performance Ratio (LSPR) workloads, which represent poor and well-scaling applications.

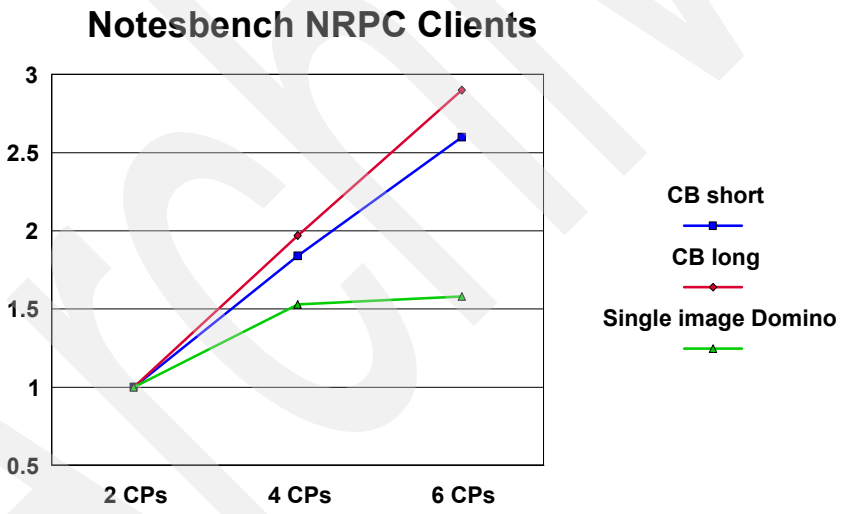



*Figure 11-3 Poor scaling application*

On zSeries, scalability is stated in terms of the transaction rate (not equivalent to Domino transaction rate!) that can be obtained with a given workload and how that transaction rate behaves when the number of processors is increased. Perfect scalability would have the transaction rate double; however, due to a number of factors, this does not happen. There are two LSPR curves that characterize workload with very good scalability (CB Long) and not very good scalability (CB short). These curves are used to indicate how an application like Domino scales. When the Linux kernel is swapping (Figure 11-3), Domino is below the CB short curve, which means it has poor scalability. When kernel swapping is removed (Figure 11-4 on page 283), Domino is very close to CB long, which means it is approaching the best workload scalability.

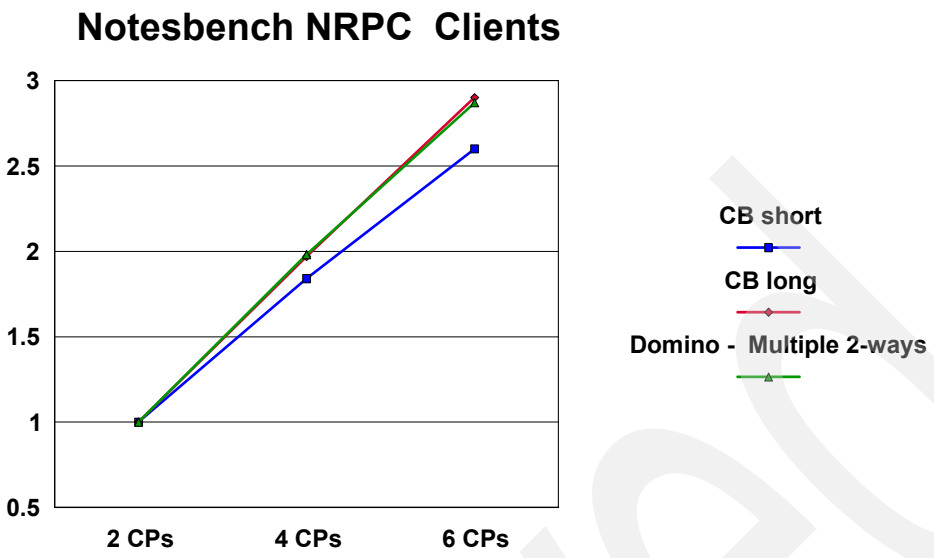



*Figure 11-4 Good scaling application*

In summary, due to the storage limitation caused by the 31-bit Linux kernel, Domino servers on a single instance of Linux will start to swap at some number of users. Although most of the scalability testing was done with LPARs, the single instance could be an LPAR image or a z/VM guest. Less CPU-intensive workload when scaled is more likely to drive swapping at fewer engines than more CPU-intensive workload. Because of swapping, more processor capacity is used to support the additional swapping as the number of users is increased on the single image. To mitigate this problem, the recommendation is to keep the total number of active users on one Linux image at just below the maximum number of users that can be supported without kernel swapping. Run Domino in multiple LPARs or z/VM guests, each with 2 to 4 CPs—2 CPs for less CPU-intensive workload, 3 to 4 CPs for more CPU-intensive workload.

## 11.4.2 z/VM processor costs

The LSC ran numerous tests to quantify the additional costs of running Notes and iNotes clients in z/VM Linux guests versus in an LPAR configuration. For Notes, 1 to 8 guests were used in the testing. The general conclusion was that the largest cost is in the initial instance of z/VM managing a single guest. The overhead of virtualizing the system resources plus the cost of providing service to a guest all have to be paid with the first guest. Subsequent guests just pay the cost of managing the services used by the guest and a small overhead for managing increased numbers of guests. The test results showed that the cost of the first guest is about 11 percent, and the cost of each additional guest, up to 8 guests, is between 1 to 2 percent. For iNotes, 1 to 16 guest scenarios were tested. The results showed that the cost of the first guest is approximately 3 percent, and the cost of each additional guest, up to 16 guests, is less than 1 percent.

The LSC uncovered an issue when running with a large number of NRPC mail users in guests under a single z/VM LPAR. It is possible to hit an I/O bottleneck in z/VM. This bottleneck is caused by the z/VM IOS module requirement that I/O controls and data reside below the 2 GB line in central storage, even though z/VM itself supports 64-bit addressing. As guests do I/O with addresses above the 2 GB line, the data and Channel Control Word (CCW) lists must be moved to storage acquired below the line, and addresses translated as appropriate. When I/O operations complete, data and controls need to be moved back into the users' addresses. This movement, acquisition of space, and address translation is

serialized. If the I/O combined rate, size and reuse is high enough, it may create a bottleneck, preventing the full use of the available processor capacity. The cause of this bottleneck is well understood, and a fix is planned for a later release of z/VM.

The Domino mail application is typically very I/O intensive. When deploying a large number of mail users, say in the thousands, two or more z/VM LPARs may be required to avoid hitting an I/O bottleneck and to remove single points of failure. The limit for the number of active users per z/VM LPAR cannot be predicted. It is dependent on the number of users, the workload executed by the users, the total number of DASD I/O operations being performed, the size of the files being read and written to, and how many times the same files are referenced, among other things. These I/O characteristics will vary from customer to customer. The initial cost of each additional z/VM LPAR on the same hardware server, including its first guest, is estimated to be around 3 percent -- not as high as the first z/VM LPAR on a hardware server. This overhead is based on current knowledge of how z/VM behaves, not on test results.

The LSC also conducted tests with z/VM guests to measure the effects of skewed/unequal versus evenly distributed workloads. Test results indicated that guests with evenly distributed workloads, like most Notesbench workloads, tend to use more CPU than guests with less equal workloads. Although Domino mail servers are typically workload-balanced, they have more processing peaks and valleys than would be seen in a Notesbench workload.

In conclusion, z/VM CPU costs are very much dependent on the number of z/VM guests as well as the workload characteristics of applications running in the guests. In general, it is better to deploy fewer guests because of the additional cost for each guest. The more CPU-intensive the workload, the lower the cost of z/VM guests. The more I/O intensive the workload, the higher the cost of z/VM, especially if multiple z/VMs must be implemented. With a high I/O rate, low CPU cost per user, the Notesbench NRPC workload has a higher z/VM cost than the less CPU-intensive iNotes workload. But Notesbench workloads do not reflect production workloads. The z/VM costs incurred by Domino production environments may be lower than what is observed with Notesbench NRPC workloads because production workloads are more CPU-intensive and more dynamic in their requests for resources.

There are many z/VM manageability advantages to deploying Domino in z/VM Linux guests. However, keep in mind that a one-guest system incurs the overhead of z/VM without its benefit of being able to optimize resources for multiple guests. If you have already paid the overhead of z/VM's first guest with another application, then the costs of deploying additional Domino guests is minimal. If you are not already running other workload in z/VM guests, consider running Linux in an LPAR for a single Domino server deployment.

### 11.4.3 Optimum DPAR configuration

Another series of tests was executed to determine the impact of running multiple Domino partitions in a single Linux image. In Figure 11-5 on page 285, the same number of Notesbench NRPC users was used to drive one, two and three DPARs. In other words, for the 2-DPAR run, half of the users were assigned to each DPAR; for the 3-DPAR run, one-third of the users were assigned to each DPAR. There was no routing of mail between Domino servers for the multi-server configurations.

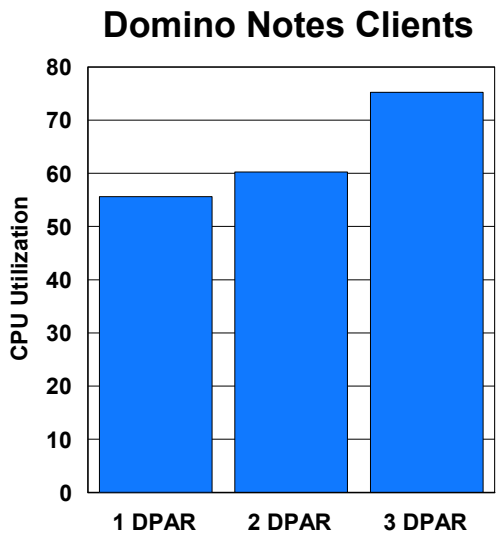



*Figure 11-5 Multiple DPARs — Notes NRPC clients*

Again, Linux kernel swapping contributed to increases in CPU for the Notes client scenario. Each DPAR in an LPAR or z/VM guest consumes a certain amount of storage. A large number of lightweight NRPC clients consume storage more rapidly than a smaller number of heavier users. In the LSC testing, these two factors caused Linux to start swapping lightly at two DPARs, resulting in a 7.6% increase in CPU as shown in Figure 11-5. Swapping was much heavier with three DPARs, resulting in a 25.6% increase in CPU.

The same tests were run with iNotes clients. Since iNotes is less CPU-intensive than Notes and consumes storage at a slower rate, Linux kernel swapping was minimal. Consequently, the DPAR CPU overhead was also quite minimal for both a second (3.3%) and third (2.2%) DPAR as depicted in Figure 11-6.

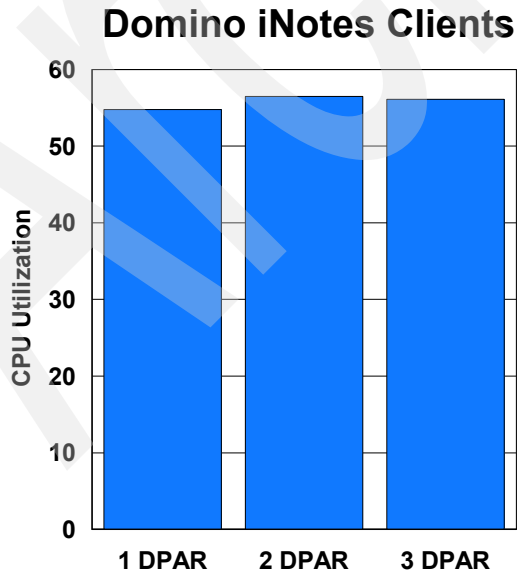



*Figure 11-6 Multiple DPARs -- iNotes clients*

Generally speaking, one fully loaded mail server per Linux image is the most efficient configuration. The more guests that are run, the higher the overhead for z/VM to manage the guests. With mail servers, there is additional CPU cost associated with more mail routing between servers; larger mail servers tend to insure that more mail is delivered locally rather than routed and then delivered. We are currently defining a fully loaded mail server as one which supports around 1000 active 15 minute mail users. For other applications, another definition of fully loaded may be more appropriate. Some Domino Linux mail customers are currently running with several hundred active users in production Domino partitions. Based on our experience with customer deployments, we believe that 1000 active users on a single mail server is a reasonable planning goal.

Although one DPAR is the most efficient configuration, this does not mean that two smaller Domino production servers would not fit into a single Linux instance. The LSC tests clearly showed that a more CPU-intensive workload, like a production workload, incurs less CPU overhead for additional servers deployed in a Linux instance. As long as kernel swapping is avoided/kept at a bare minimum and there are no issues with CPU utilization, two DPARs should fit into a single production Linux instance. However, until Domino supports a 64-bit Linux kernel, we do not recommend implementing more than two DPARs in a single Linux instance.

## 11.5 Storage planning

Domino 6.5.x releases are currently only supported in a 31-bit Linux kernel, which means that each Linux image is limited to 2 GB of central storage and some amount of storage for Linux swapping. As discussed earlier, this limitation will go away with Domino 7 when 64-bit Linux becomes a supported environment. Unlike some other platforms, Linux is architected to exploit all of the central storage in an OS image, claiming any available storage not used by applications as part of its file cache. If 2 GB of central storage is defined to a Domino Linux image, all 2 GB will be marked as utilized whether or not Domino is actually using all of that storage.

The fact that Linux allocates excess storage to its file cache makes planning for storage somewhat challenging. It is important not to define too much storage, resulting in a significant portion of it being wasted when populated with file cache. Yet, there must be enough storage so that Linux swapping is minimized. 11.9.6, “Reducing Linux cache size” on page 301 illustrates an exercise in reducing Linux file cache size.

Some amount of storage, even if minimal, should always be defined as Linux swap space, either as XPRAM (expanded memory) or optionally as a z/VM virtual disk when running in a z/VM guest. See 11.10.2, “Using virtual disk for swap” on page 305 and 11.10.3, “Virtual storage versus virtual disk swapping trade-off” on page 307 for an in-depth discussion of using virtual disk for Linux swap space. Although Linux adjusts the size of its file cache as needed, returning storage to applications as their demands for it increase, this process is not dynamic enough to totally avoid swapping. The LSC’s tests showed cases where Domino Linux instances had plenty of storage, much of it allocated to file cache, yet some Linux kernel swapping still took place as Domino’s storage demands increased.

There are numerous factors to consider when planning for Domino storage requirements, among them:

- The number of connected NRPC clients, including Notes and DAMO clients

  Domino Access for Microsoft Outlook (DAMO) clients use the NRPC protocol for communicating with Domino backend servers.

- The number of active browser mail clients, including DWA, Webmail, POP3, and IMAP clients
- The number of Domino partitions
- The number and types of server tasks running in each Domino partition
- The number of Linux images
- The number of z/VM LPARs if running z/VM guests

Some of the factors listed above may be difficult to project ahead of time. So, we offer some general guidelines for Domino storage planning.

If you run Domino in an LPAR:

- Define up to 2 GB of central storage

  You may be able to run with considerably less than 2 GB of storage, especially if only a single DPAR is deployed in the LPAR. A larger than needed amount of central storage has no performance implications. However, allocating storage beyond what is needed for the application will allow the Linux file cache to grow, without necessarily improving throughput. If the application requires less than 2 GB, adjustments may be made to decrease central storage, but only to a point where there is no Linux swapping.
- Define up to 1 GB of expanded storage for swapping

  Although more expanded storage can be defined, testing has shown that Linux will not exploit anything beyond 1 GB.

If you run Domino in a z/VM guest:

- Define 1 GB of central storage as a starting point
- Define 256 MB of storage for swapping as a starting point
- Increase/decrease the amount of central storage to some point so that the Linux kernel is not swapping

  Increasing storage above 1 GB may cause some z/VM paging. However, be aware that z/VM is more efficient at paging than Linux is at swapping. So, when there is a choice to be made, it would be better to give Linux guests as much virtual storage as they require to avoid swapping, and let z/VM do its paging.
- To start with, do not over-commit storage by a factor greater than 1.5 or 2

  The term over-committed storage means how much storage has been defined as compared to how much is available. In z/VM, it is a way to enforce resource sharing. If the total amount of storage in the working set of all the guests exceeds the amount of real storage assigned to the z/VM LPAR, then z/VM will page virtual storage to expanded storage on the paging device. This paging can be costly if storage is over-committed. For Domino, it is very important not to over-commit storage to the point of not having enough storage for active Domino guests. For a discussion of the z/VM storage hierarchy, see 11.10.1, “Storage hierarchy” on page 304.
- Plan for 500 MB of storage for each z/VM LPAR

## 11.6 Network planning

A key area when doing capacity planning is to examine the network.

## 11.6.1 Configuration

One of the key performance factors in any deployment is the way that the network is configured. There are a number of options one could choose for the setup. Figure 11-7 shows the network setup for the LSC's Domino Linux LPAR tests, where a single instance of Domino is running on a Linux kernel in an LPAR. Using this configuration, multiple Domino partitions can be created to share the same Domino instance or set of executables. Each DPAR has its own set of processes where some number of users will run.

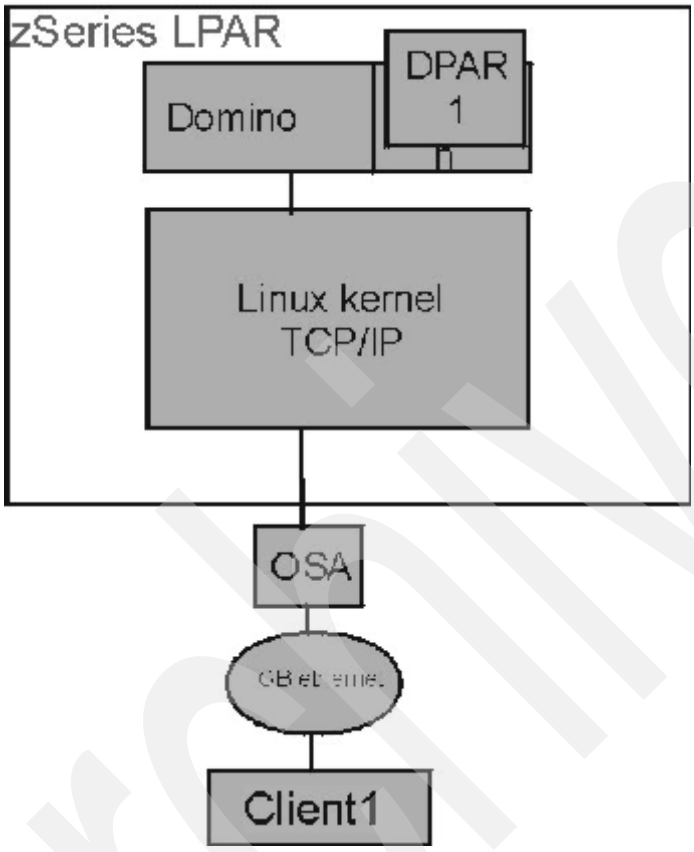



*Figure 11-7 Single DPAR*

The Linux kernel is directly connected to an OSA card through addresses which are dedicated to this kernel. An OSA card has up to 80 sets of addresses which can be dedicated to a system. In Figure 11-7, only one of the 80 sets of addresses is in use, and is dedicated to the LPAR. This is the best performing network configuration because it has no additional overhead. However, this configuration requires that the network administrator set up the direct connections to each kernel using the OSA card.

Figure 11-8 on page 289 shows the configuration where an OSA card is directly connected to a single z/VM guest. With this setup, z/VM starts I/O operations and handles ending interrupts from the OSA card, and passes control to the owning guest. One of the OSA connections is dedicated to this single guest. z/VM imposes some overhead, but provides some administrative simplification in the definition and activation of multiple guests that are not provided when implementing multiple LPARs.

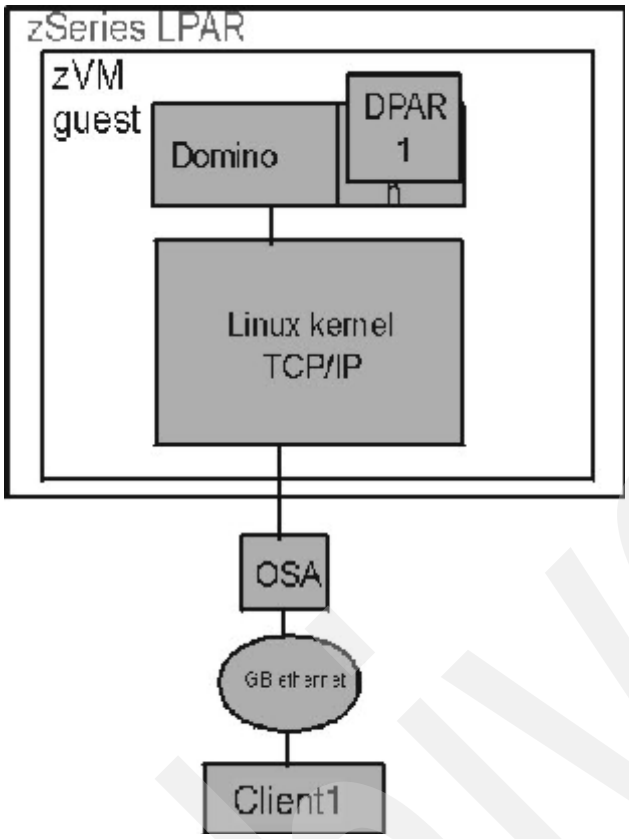



*Figure 11-8 DPAR in single z/VM guest*

Figure 11-9 on page 290 shows the configuration which was used for the LSC's tests with multiple z/VM guests. Each z/VM guest has a dedicated connection to the OSA card. The LSC, which tested scenarios using up to 16 guests, only needed one OSA card to handle all of the network traffic generated during the benchmarks. In a production environment, customers may want to use multiple OSA cards if the throughput is above what the OSA card can support. If multiple OSA cards are deployed, consider using virtual IP addresses (VIPAs) to allow load balancing of the network traffic.

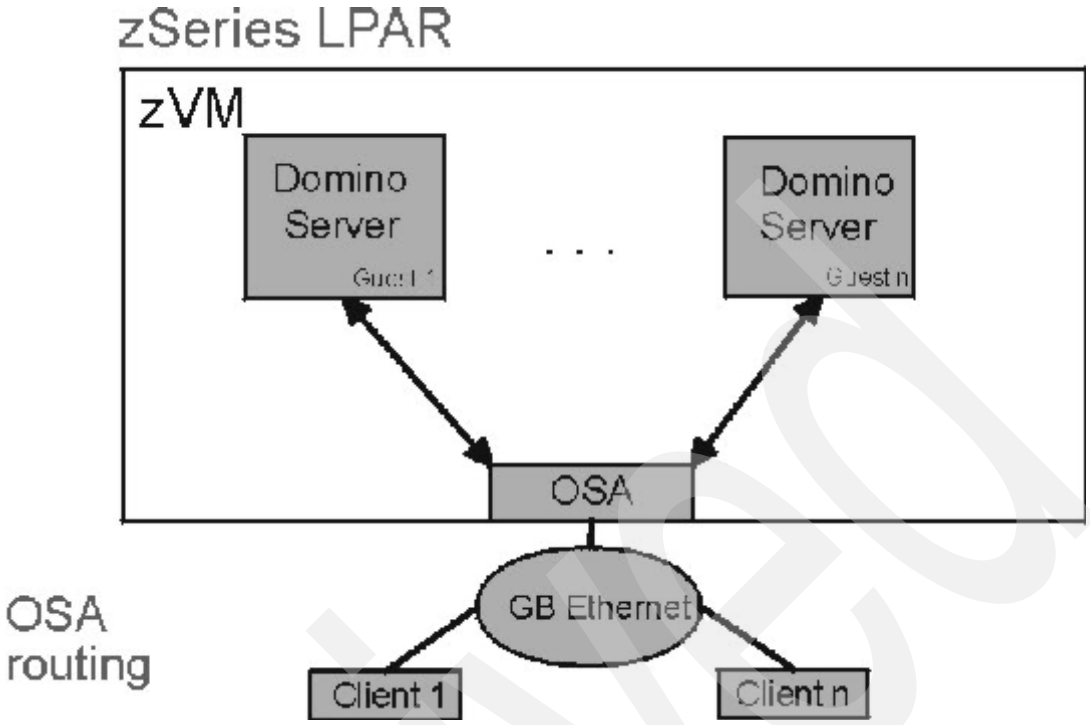



*Figure 11-9 Multiple z/VM guests*

Figure 11-10 on page 291 shows an alternative configuration where multiple z/VM guests are connected to an OSA through a virtual switch, which was new in z/VM V4.4. With this setup, z/VM provides the capability to have a single OSA connection shared by a number of guests through the virtual switch. Guests connect to a z/VM guest LAN, and the virtual switch function does the routing of the messages to the correct guest. For this configuration, there is additional overhead to provide the routing function. However, there is less overhead with a virtual switch than there is with a guest system doing network routing. For more information about implementing a virtual switch, see the Redpaper: *Linux on IBM @server zSeries and S/390: VSWITCH and VLAN Features of z/VM 4.4,* REDP-3719.

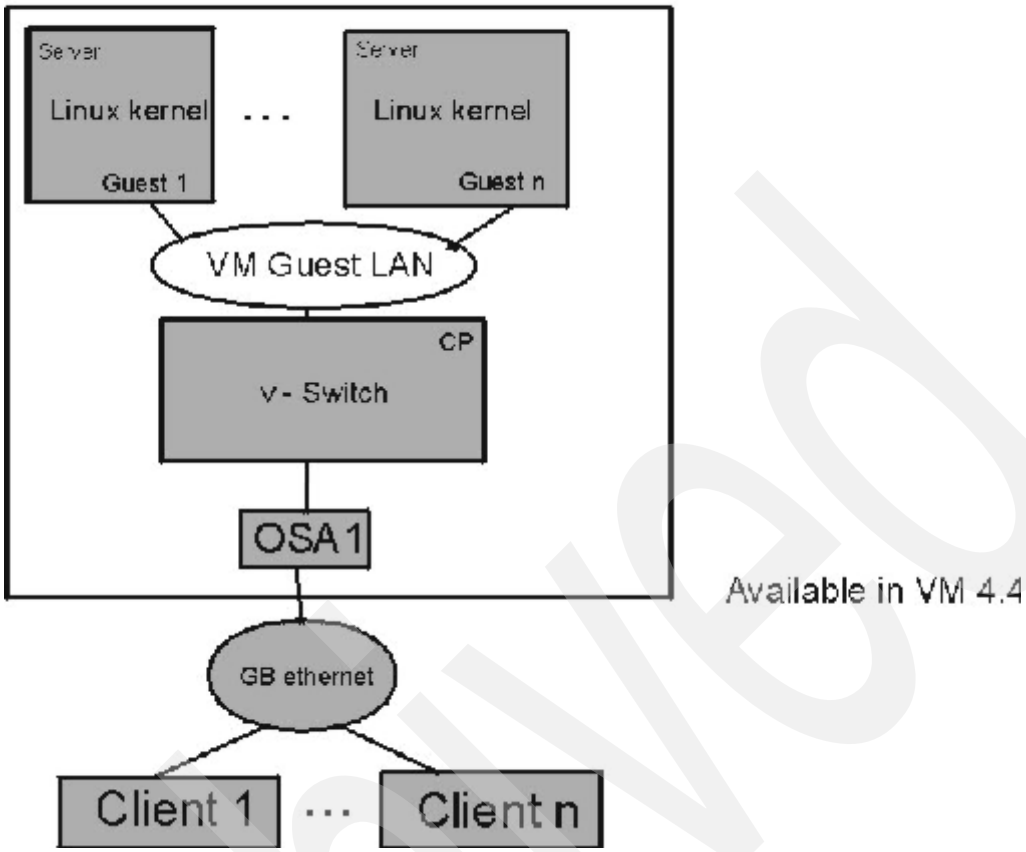



*Figure 11-10 Multiple guests with virtual switch*

## 11.6.2 Domino considerations

When consolidating distributed environments to a central location, end user response time issues may arise due to network constraints. Domino has a few ways to address constrained networks, the first being deployment of local replicas for mail/application files on user workstations. Working off of local replicas reduces network traffic to Domino servers and off loads more workload from servers, potentially lowering CPU as well as bandwidth requirements. For the Domino mail application, we also recommend putting a mobile directory on workstations. Local directories allow end users to perform type-ahead addressing without accessing Domino servers, again reducing network traffic and CPU. No formal testing has been performed to measure network bandwidth or CPU reductions when local replicas and mobile directories have been deployed on workstations. However, we have received feedback from customers who have experienced improved end-user response times as well as flatter CPU peaks with the implementation of local replicas. There have been a few customer situations where CPU requirements increased with local replicas on workstations. This increase was found to be related to the frequency of replication with Domino servers. For a large number of users, a replication interval of 5 minutes or less can drive up CPU utilization substantially. Our recommendation is to replicate no more than every 10 minutes.

Network compression is an important performance feature first offered in Lotus Notes/Domino 6. When enabled, this feature automatically compresses data before it is sent over the network, improving network performance over slower line speeds. It is available for server to server traffic and client to server traffic. For compression to work, it must be enabled on both sides of a network connection. To enable compression for a network port on a server, use the Server tab in the Domino Administrator. To enable compression on network ports on

Notes workstations, use a setup or desktop policy settings document from the Domino Administrator to push down the appropriate settings to the workstation. Network compression will reduce bandwidth costs, but will also increase CPU utilization. The increases in CPU utilization are difficult to quantify because they are related to the amount of network data being compressed. To date, no formal studies have been done to correlate network traffic sizes/rates to network compression CPU costs.

For browser access, the secure sockets layer (SSL) may be an important security feature. Unfortunately, establishing the SSL session or handshake and the encryption of the data can be quite costly. Tests performed in the past have shown the CPU overhead to be as high as 30%. The zSeries cryptographic hardware reduces some of that overhead. However, Domino 6.5 on Linux does not directly support this hardware. As a circumvention, there is the option to put an Apache proxy server to access the cryptographic hardware in front of Domino.

## 11.7 Disk/DASD planning

Another key area for capacity planning is the DASD configuration

### 11.7.1 Amount of disk

The disk requirements for a base Domino 6.5 server are documented in Chapter Chapter 4, “Disk configuration” on page 53 of this redbook. With new Domino releases, disk requirements often change. For example, disk requirements for the Domino executables typically increase to support improved functionality within Domino. However, most of Domino’s disk requirement is directly tied to application requirements:

- Number and average size of mail files
- Number and size of other application databases
- Whether Domino clustering is implemented

  Domino clustering supports failover of users to alternate servers when the server they are working on becomes unavailable. Since Domino is not architected to share files between servers, this means that multiple replicas of the same databases must exist in two or more servers, easily doubling disk requirements.

### 11.7.2 Placement of Domino databases

Domino is a very disk I/O intensive application. It is important that all the Domino system and user files be spread to optimize DASD response time for the Domino volumes. If response times are not optimal, performance will be adversely affected.

### 11.7.3 ESS default cache mode

The DASD control unit default cache mode in SuSE SLES 8 SP2 and below is not correct for the Domino workload. The default cache mode for the Enterprise Storage Server® (ESS) DASD control unit in ECKD™ mode is ‘cache sequential’. This proves to be much less efficient for Domino than the ‘cache normal’ or random mode. For SLES 8 SP2 and below, the default should be changed to cache normal. For SuSE SLES8 SP3 or above, the default is already set to cache normal. SuSE SLES8 SP3 is the recommended maintenance level for Domino.

## 11.8 Other Domino impacts to capacity

Here are additional considerations when doing capacity planning for Domino.

### 11.8.1 Clustering

Domino clustering provides failover and load balancing capability for Domino users. Multiple replicas of users' databases are created on two or more Domino servers. Database changes are synchronized across all replicas almost immediately through cluster replication. When a server fails or is too busy, users are directed to other servers with up-to-date replicas of databases that they are trying to access. As mentioned in 11.7, "Disk/DASD planning" on page 292, disk costs go up because of multiple copies of databases.

But more importantly, CPU and storage requirements increase. For two-cluster environments (two Domino servers with two sets of replicas), there are twice as many Domino partitions, and potentially twice as many Linux images. As described in previous sections, additional DPARs and Linux images incur some CPU overhead and add storage costs. However, the largest portion of the cost is the synchronization of the clustered servers' databases. CPU costs are directly tied to the number of times the same updates must be applied to multiple databases. In our experience, this overhead could easily be anywhere from 30 to 50%.

To reduce the costs of Domino clustering:

- Limit the number of users who are clustered

  Only cluster those users with a real need for failover. Casual users whose day-to-day activities are not dependent on e-mail or other applications probably do not have a need for the highest availability.

- Deploy only two-way clusters

  Clusters with more than two servers incur even higher capacity costs because of more redundancy of DPARs, Linux images and database replicas.

### 11.8.2 Agents

Agents are stand-alone programs which perform a specific task in one or more Domino databases. These programs are executed by the Domino Agent Manager task. They can be triggered by an event, such as the arrival of new mail in the inbox, or they can be scheduled at a specific frequency per day, week, or month. Agents can run on client workstations or on Domino servers. From a capacity perspective, we are only concerned with agents which run on Domino servers. To reduce the CPU utilization of the Agent Manager task:

- Run system/administrative agents which process many databases during periods of low end-user activity
- Disable personal agents for end-users.

  Giving users the ability to run personal agents on Domino servers is giving them a blank check to consume CPU resources. Personal agents used to perform functions like managing an inbox can be very costly if executed by a large user community. For those users who require personal agents, a better solution is to run the agents on a workstation against a local replica of the database(s). Personal agents for the general user population can be restricted through the Programmability Restrictions section in the Domino Server document.

### 11.8.3 Full-text indexing

Full-text indexes allow users and programs to quickly search for information in databases. Unfortunately, there is CPU overhead associated with building/updating the indexes and a even greater overhead in performing searches againsts indexes. From a capacity point-of-view, it is important to choose which databases really require full-text indexes. For example, the costs associated with full-text indexing 50% of all mail files in a company may not be financially prudent. If using Notes clients, users have the option of creating and full-text indexing local replicas. To reduce capacity requirements, some company's limit or totally disallow full-text indexing on Domino servers. Full-text indexing can be turned off on Domino servers by setting the notes.ini parameter Update_No_Fulltext to '1'.

### 11.8.4 Special functions and third-party products

The capacity requirements of certain functions and features in Domino cannot be ignored. For example, Message Tracking, which tracks delivery of all mail messages, has been observed to utilize a large amount of CPU when enabled. Third-party tools such as anti-virus have associated costs. Sometimes, a specific option enabled for features or tools is a capacity consideration. For example, an anti-virus tool which has been enabled to scan only incoming mail impacts capacity considerably less than one which scans all databases.

### 11.8.5 Notes.ini parameters

Several notes.ini parameters can be set to reduce capacity requirements:

#### IOCP_DISABLE_ASYNC_NOTIFICATION=1

The Domino mail server by default checks every connected user once per minute for new mail and then notifies those users with new mail. Setting the above parameter to '1' disables this functionality, reducing CPU utilization. The frequency of the check may be changed to some value greater than one minute with the notes.ini parameter IOCP_Async_Notification_Poll_Time. This function should not be confused with client polling for new mail, which is set in the User Preferences of the Notes client.

#### FT_FLY_INDEX_OFF=1

Domino allows searches by agents against databases without full-text indexes. If full-text indexing is not disallowed on the server, it will build these indexes dynamically each time an agent invokes a search, causing significant CPU spikes. There are two options. One is to build the indexes ahead of time to avoid the dynamic indexing. The other is to set this parameter to '1' to disallow the building of dynamic indexes.

## 11.9 Capacity monitoring

Capacity planning is the function of projecting resource requirements for a workload or an installation to assure that sufficient resources are available to perform required work; in traditional mainframe environments, being a capacity planner is often a very important career position in an IT organization.

However, when servers began being implemented on low-cost platforms instead of on mainframes, it became less expensive to buy a new server rather than spend the time and resource for tuning. Upgrades did not have to be planned months or years in advance, but in days or weeks.

Now, with the consolidation technique of moving smaller servers back to the mainframe, capacity planning is again a requirement. Sometimes in the pressure of the moment, we are tempted to “guess and go”. A better approach is to collect performance data for a period of time, analyze the requirements, and then use that data to calculate requirements.

Projecting capacity requirements from an Intel-based server to a mainframe can be done. Using the “4” megaHertz-to-MIP estimator (Barton’s number, as presented at SHARE and the IBM Technical Conference[a]) is one methodology. This methodology takes the resource requirements on Intel in terms of MHz, divides that by 4, and that provides the MIP requirement. This is not exact, but is easily duplicated by any installation testing their consolidation plan. For example, if an application required 20% of a 1 GHz Intel processor, then it would take about 50 “MIPS” of a mainframe. This should provide a conservative estimate in size of processor required.

Each subsystem can be evaluated using a similar process. Using this methodology, installations have the ability to size existing workloads and build their own capacity model based on their experiences.

a. Barton Robinson, one of the redbook authors, is the founder of Velocity Software. See 12.7, “ESALPS overview with guest under z/VM” on page 330 for a brief description of their toolkit.

### 11.9.1 Performance reporting

In preparation for understanding capacity requirements, real data is required. To provide examples of how to collect this data, and how to interpret this data, NETSNMP was used with the NotesBench driver, with the reporting provided by Velocity Software’s ESALPS. Although other tools are available to measure capacity and performance, for purposes of illustration, ESALPS was used as part of the initial redbook project. All report examples in this chapter are produced by ESALPS.

Velocity Software staff participated in this redbook project because of their knowledge of z/VM performance, Linux consolidation practices, and Linux skills. ESALPS was used to collect the performance data, as well as for analysis and presentation; see 12.7, “ESALPS overview with guest under z/VM” on page 330 for more details about this tool.

NETSNMP (from sourceforge.org) was run on three Linux servers that had Domino installed. Data was collected using ESALPS to create a set of history files from which this data was reported. This history data includes VM data and Linux data with a one-minute granularity.

### 11.9.2 Infrastructure issues

One reason to do capacity planning is to purchase only the IT infrastructure that is required to run your business. The cost of infrastructure is less important if the number of virtual servers is small. From a performance data collection perspective, there are agents used to collect data. These agents are unrelated to Domino, and there are many of these agents available to support different aspects of systems management.

If the cost of running performance (or other systems management) agents in each Linux is 2% of a processor, and this is only multiplied one or two times, it is a trivial expense. However, if there will be one hundred servers, then the cost at 2% each requires two processors just for the performance agents. Any method for performance analysis or capacity planning should be evaluated for this shared environment prior to the solidifying of the configuration.

In an environment where resources will be shared by two or more virtual servers, any resources used by one process reduces the amount of resources available to other servers. The cost of running non-required or inefficient processes thus reduces the overall workload

capability of the system. All additional processes should be evaluated for suitability. For example, if you find cron jobs that start regularly but are not always necessary, this is work that is easily eliminated.

When measuring inside the Linux server, an agent is required to provide the data. Though “top” is the most common performance monitor in a single server dedicated environment, the resource requirements of top are not suitable to a shared resource environment.

There are two types of agents, passive and active:

- An active agent is constantly waking up, collecting data, writing it to a log file, and then going to sleep. This may occur as often as every two seconds. The data collection is done on the server, and some additional work is required to move that data from the server to where the data will be analyzed.
- A passive agent—netsnmp is the example used in this book—sits idle until there is an external request for data. The result is that control over data collection is an external function rather than an internal one. Also, the data now resides in a central repository for all servers.

### 11.9.3 Linux measurement inaccuracy

When an agent inside Linux under z/VM performs CPU measurements, Linux assumes and reports the data based on 100% of the machine. This leads to sometimes very exaggerated values.

In Figure 11-11 on page 297, each Linux guest was measured twice. This data is provided by Linux. The reported value by Linux was LinuxA2, LinuxB2, and LinuxC2. The true (corrected) values are reported as LinuxA, LinuxB, and LinuxC. Thus LinuxA2 and LinuxA are reporting on the same Linux server. The external data collector (as part of ESALPS) corrected the LinuxA value based on the z/VM correct data. The data provided by z/VM for each virtual machine is correct to nearly a microsecond.

**Note:** This same reporting inaccuracy also happens when Linux is running natively in an LPAR. As noted above, z/VM has the benefit of providing correct reporting. If running both Linux and z/OS LPARs on the same hardware server, accurate reporting of CPU utilization for Linux LPARs can be obtained through SMF data collected for all LPARs in a z/OS LPAR. If the CPs are dedicated to the Linux LPAR, the Linux CPU utilization reporting should be accurate.

Performing capacity planning using invalid numbers can be a very serious mistake. On heavily loaded systems, Linux has been shown to report an order of magnitude increase in CPU requirements over what is actually utilized. When measuring dedicated servers, this is not an issue. Thus, on servers where processors are dedicated, or even shared but with very low utilization, this is not an issue. It is an issue whether running under z/VM or in an LPAR on a shared system, but is easier to correct under z/VM.

```
Screen: ESAUCD4  ITSO z/Domino Redbook
1 of 2  LINUX UCD System Statistics Report

                  <Processor Pct Util> NICE
Time     Node     Total Syst User Idle Time
-------- -------- ----- ---- ---- ---- ----
13:24:00 LINUXC2   28.1 12.6 15.5  171  0.0
         LINUXB2   28.1 15.0 13.1  171  0.0
         LINUXA2   23.1 10.3 12.8  177  0.0 ==> uncorrected Linux data provided by Linux
         LINUXB    19.9 10.6  9.3  170  0.0
         LINUXC    29.8 14.2 15.6  169  0.0
         LINUXA    18.1  8.6  9.6  174  0.0 ==> Linux data corrected by ESALPS
13:23:00 LINUXC2   31.5 14.6 16.9  168  0.0
         LINUXB2   30.4 17.0 13.4  169  0.0
         LINUXA2   34.8 18.3 16.5  165  0.0
         LINUXB    19.2 10.0  9.3  173  0.0
         LINUXC    30.6 14.0 16.6  170  0.0
         LINUXA    17.9  8.3  9.6  173  0.0
```

*Figure 11-11 ESAUCD4 Linux server processor reporting - showing accuracy problem*

## 11.9.4 Profile of a Domino server

Figure 11-12 on page 298 shows the processes of interest in each Linux. The Java, router, and server processes were replicated many times. This report summarizes, for each server, the number of processes, the number of active processes (those using some amount of CPU during a one minute interval), the number of running processes, and the number in resource wait. The total percentage of processor used by each process group is provided with CPU seconds and average CPU per process.

This summary was found useful in understanding how the CPU requirement was distributed. Two processes of interest were kswapd, where the overhead for swapping was assigned, and the update process, which in many cases was the largest consumer of CPU resource.

```
 Report: ESAHSTA    LINUX HOST Application Report   Domino Redbook    ESAMAP 3.4.0 08/25/03  Page 141
 Monitor initialized:                 on 2066 serial 71CE3  First record analyzed: 08/24/03 18:00:00
-----------------------------------------------------------------------------------------------------
 Node/    Process/    <-Application Process Counts------> <-----Processor---->
 Date     Application                                     <---Utilization---->
 Time     name        Total active Running ResWait Loaded Percent seconds Avg
 -------- ----------- ----- ------ ------- ------- ------ ------- ------- ----
 19:00:00
 LINUXA                                                                          One DPAR / Linux Server
          event         6.0    6.0     2.0     4.0      0     0.7    25.3  0.1
          java         14.0   14.0     1.0    13.0      0     9.1   327.5  0.6
          ksoftirq      2.0    2.0     1.0     1.0      0     1.5    54.7  0.8
          kswapd        1.0    1.0       0     1.0      0     0.7    25.4  0.7
          kupdated      1.0    1.0       0     1.0      0     0.0     0.3  0.0
          router       13.0   13.0     3.0    10.0      0     2.8    99.1  0.2
          sched         5.0    5.0     1.0     4.0      0     0.4    14.7  0.1
          server       68.0   68.0     2.0    66.0      0    11.3   405.6  0.2
          snmpd         1.0    1.0     1.0       0      0     2.6    92.2  2.6
          update        6.0    6.0     1.0     5.0      0     9.2   332.5  1.5
LINUXC                                                                         Two DPARs in one server
          event        13.0   13.0     3.0    10.0      0     1.2    44.4  0.1
          iostat        4.0    4.0       0     3.0      0     0.0     0.2  0.0
          java         32.0   32.0       0    32.0      0    15.5   557.3  0.5
          ksoftirq      2.0    2.0     1.0     1.0      0     2.8   101.4  1.4
          kswapd        1.0    1.0       0     1.0      0     3.4   124.1  3.4
          router       32.0   32.0     3.0    29.0      0     5.7   203.9  0.2
          sched         9.0    9.0       0     9.0      0     0.1     4.3  0.0
          server      137.0  137.0    10.0   123.0    4.0    24.0   864.5  0.2
          snmpd         1.0    1.0     1.0       0      0     4.5   161.0  4.5
          update       11.0   11.0       0    11.0      0    18.2   656.5  1.7
LINUXB                                                                         One DPAR in one server
          event        10.0   10.0     2.0     8.0      0     0.8    27.3  0.1
          java         15.0   15.0     1.0    14.0      0     8.3   299.2  0.6
          ksoftirq      2.0    2.0     2.0       0      0     1.9    67.4  0.9
          kswapd        1.0    1.0       0     1.0      0     0.6    20.8  0.6
          router       13.0   13.0     3.0    10.0      0     2.9   104.5  0.2
          sched         5.0    5.0     1.0     4.0      0     0.4    13.0  0.1
          server       66.0   66.0     4.0    60.0    2.0    10.5   378.8  0.2
          snmpd         1.0    1.0     1.0       0      0     2.3    81.9  2.3
          update        5.0    5.0     1.0     4.0      0     6.9   250.1  1.4
```

*Figure 11-12 ESAHSTA Linux application profile*

## 11.9.5 Processor analysis

Processor capacity planning should begin by analyzing the requirements of the workload. Ideally, performance would be gathered for several weeks or months to understand when the workload peaks and how much resource is required during those peaks.The data that can be collected depends on what agents are installed. Using the SNMP interface provided by ESALPS, there are two types of data that can be acquired: processor data and process data.

Figure 11-13 contains an ESAMON display showing five minutes of processor data provided by a Linux server running the NETSNMP implementation.

```
Report: ESAUCD4      LINUX UCD System Statistics Report            ITSO
Monitor initialized:                      on 2066 serial 71CE3     First
----------------------------------------------------------------------
Node/                                <---------Rates Per Second------>  <
Date/    <Processor Pct Util> NICE  <-Swaps-> <-Blocks> Switch Intrpt  <
Time     Total Syst User Idle Time    In  Out   In  Out   Rate   Rate  1
-------- ----- ---- ---- ---- ----  ---- ---- ---- ---- ------ ------  -
08:31:00
LINUXA   130.0 82.0 48.0 69.1    0  24.0  8.0  111  348  158.0      0
----------------------------------------------------------------------
08:32:00
LINUXA   128.5 80.2 48.3 70.4    0  24.0  8.0  111  348  158.0      0
----------------------------------------------------------------------
08:33:00
LINUXA   104.1 58.3 45.8 91.2    0  24.0  8.0  111  348  159.0      0
----------------------------------------------------------------------
08:34:00
LINUXA   109.3 60.4 48.9 95.2    0  24.0  7.0  111  347  160.0      0
----------------------------------------------------------------------
08:35:00
LINUXA   104.6 56.5 48.1 90.3    0  24.0  7.0  111  347  160.0      0
```

*Figure 11-13 ESAUCD4 Linux processor utilization example*

Figure 11-14 shows the process data for one of those one-minute intervals.

```
Report: ESAHST1      LINUX HOST Software Analysis Report
Monitor initialized:                      on 2066 serial 71CE3
-------------------------------------------------------------------
Node/    <-----Software Program-------> <CPU Seconds> CPU  StgSize
Time     Name      ID    Type   Status  Total Intrval Pct  (Bytes)
-------- --------  ----  ------ ------- ----- ------- ---- -------
08:32:00
LINUXA
         init         1  Applic ResWait   0.9     0.0  0.0   61440
         kjournal    95  Applic ResWait   2.5     0.0  0.0       0
         db2fmd     596  Applic ResWait   0.3     0.0  0.0  573440
         sshd      1081  Applic ResWait   0.4     0.0  0.0  204800
         event    10787  Applic ResWait  19.5     0.0  0.0  11188K
         snmpd    10861  Applic Running 193.4     4.2  7.1   1492K
         adminp   11452  Applic ResWait  58.5     0.0  0.1  13848K
         server   11525  Applic ResWait   1.0     0.1  0.1  35720K
         server   11533  Applic ResWait   4.3     0.0  0.0  35720K
         server   11537  Applic Running 44697    58.3 99.2  35720K
         java     13024  Applic ResWait   0.0     0.0  0.0   6632K
         java     24016  Applic ResWait   1.9     0.0  0.0   6632K
         java     24024  Applic ResWait   4.9     0.0  0.0   6632K
         server   24192  Applic ResWait  19.0     0.1  0.1  35720K
         java     26352  Applic ResWait   0.4     0.0  0.0   7320K
         sshd     26477  Applic ResWait   0.2     0.0  0.1   2028K
```

*Figure 11-14 ESAHST1 Linux process utilization example*

One other perspective, useful for capacity planning, is the ability to look at all the processes as a group. Looking at a sorted list of several hundred processes provides only a quick idea of which processes are using CPU. The ESAHSTA report (in this case, from ESAMAP) groups

all of the processes of the same name and reports as a group. Record how much of the resource is taken by Java (or update or server), then you can characterize your workload.

```
Report: ESAHSTA     LINUX HOST Application Report    Domino Redbook ESAMAP 3.4.0
Monitor initialized: on 2066 serial 71CE3     First record analyzed: 08/24/03 18:00:00
-------------------------------------------------------------------------------------
Node/    Process/    <-Application Process Counts------> <-----Processor---->
Date     Application                                     <---Utilization---->
Time     name        Total active Running ResWait Loaded Percent seconds Avg
 -------- ----------- ----- ------ ------- ------- ------ ------- ------- ----
LINUXC
          event        13.0   13.0     3.0    10.0      0     1.2    44.4  0.1
          iostat        4.0    4.0       0     3.0      0     0.0     0.2  0.0
          java         32.0   32.0       0    32.0      0    15.5   557.3  0.5
          ksoftirq      2.0    2.0     1.0     1.0      0     2.8   101.4  1.4
          kswapd        1.0    1.0       0     1.0      0     3.4   124.1  3.4
          router       32.0   32.0     3.0    29.0      0     5.7   203.9  0.2
          server      137.0  137.0    10.0   123.0    4.0    24.0   864.5  0.2
          update       11.0   11.0       0    11.0      0    18.2   656.5  1.7
```

*Figure 11-15 ESAHSTA Linux application example*

## Processor planning

Processor planning can sometimes be done by guessing at the number of users that will be supported and the amount of resource used by those users. But since every company (maybe every department) will have a different work profile, guessing is just guessing. For installations that run a Domino workload already, the best method would be to capture the performance data for a month to determine existing processor requirements.

If the data from the above ESAHSTA report was captured for a month from one platform, it is then possible to estimate CPU requirements on another platform. For example, if the total requirement is an average of 10% of a 1 GHz Intel-based processor, you could estimate an average requirement of about 25 MIPS requirement on a z800 or z900. This calculation is an estimate based on “Barton’s Number” of 4, which calculates the MHz requirement, in this case 100 MHz, and divides by 4 to get an estimate of MIPs requirement.

One of the functions of ESALPS is to collect data from many other platforms. Using either NETSNMP or another SNMP implementation, ESALPS will collect the performance data and store it into the Performance DataBase. This data is then used to project future capacity requirements.

## Idle servers

When running large numbers of servers, there is a cost that should be recognized. An idle server seems to require some amount of storage and processor resources. Factors include the number of virtual processors per virtual server, and the configuration of the network. The cost of this for us seemed to be about 2 to 3% of a z800 processor. Thus, one hundred idle servers has the cost of at least two z800 engines. Some consolidation will help reduce the infrastructure and management cost of this environment.

Although we did not trace this, possible causes for this CPU usage include:

- We were running with the Linux timer enabled and at the default setting of 100 times per second. This keeps Linux active and ineligible to page. We recommend that you apply a kernel timer patch (see 10.2, “Idle server still uses 2 to 3 percent of system” on page 268) to reduce the number of timer pops.

Also examine APAR VM63282 to see if your situation calls for it. It is discussed in 7.4, “QDIO and the dispatch queue” of *Linux on IBM @server zSeries and S/390: Performance Measurement and Tuning,* SG24-6926.

**Tip:** Authors of the SG24-6926 redbook believe that a shared kernel is advantageous when running Linux under z/VM. They explain how to do this in 4.2, “Exploiting the shared kernel” of that redbook.

- Domino servers with no workload actually consume some amount of CPU through numerous processes and threads performing sleep() /usleep() calls, only to wake up find out there is no work to do and go back into sleep() /usleep().

## 11.9.6 Reducing Linux cache size

In a virtualized environment where there is significant benefit in sharing resource, reducing storage requirements for each server benefits the total system capacity. When too much storage is provided to Linux, a Linux server will populate that storage with cached data. This can be a large misuse of valuable storage resource. To reduce this storage, Linux memory can be reduced, and Linux will in turn reduce the size of its cache.

One of the experiments that we ran during the project was to start with a smaller virtual machine. Figure 11-16 shows the storage analysis for a run with a large number of users.

```
Report: ESAUCD2      LINUX UCD Memory Analysis Report              ITSO
Monitor initialized:                      on 2066 serial 71CE3     First record
-------------------------------------------------------------------------------
Node/    <-------------------------Storage Sizes (in MegaBytes)------------->
Time/    <--Real Storage--> <-----SWAP Storage----> Total <--Storage in Use->
Date     Total  Avail Used  Total Avail Used  MIN   Avail Shared Buffer Cache
-------- ------ ----- ----- ----- ----- ----- ----- ----- ------ ------ -----
08:01:00
LINUXA    249.2   3.0 246.2 242.0  94.3 147.6  15.6  97.4      0    5.0 192.4
-------------------------------------------------------------------------------
08:02:00
LINUXA    249.2   2.9 246.4 242.0  94.3 147.6  15.6  97.2      0    5.1 192.5
-------------------------------------------------------------------------------
08:03:00
LINUXA    249.2   2.8 246.5 242.0  94.3 147.6  15.6  97.1      0    5.2 192.5
-------------------------------------------------------------------------------
```

*Figure 11-16 ESAUCD2 Linux memory analysis, 256 MB run*

The important numbers from this are:

**Total real storage** The virtual machine size was 256 MB; Linux reports 249.2 MB of storage, reserving about 5% for the kernel, and of which 246.2 was in use. But this is not actually meaningful, as Linux will use all available storage for cache when available.

**Swap storage** With a total of 242 MB available, 147 was in use.

**Cache storage in use** 192.4 MM. This is the main use of storage in this example.

Recognizing that the cache size was much larger than required, we reduced the virtual machine size to 196 MB. Figure 11-17 on page 302 shows that the swap was used slightly more, going from 147 MB to 165 MB, but more importantly, the cache size dropped

significantly. The overall requirements of z/VM storage dropped from a total of about 400MB (256 + 147) to 360MB (196 + 165).

For this workload, no degradation in response time was measured when reducing storage. CPU usage of Linux and VM should be monitored to minimize high rates.

```
Report: ESAUCD2      LINUX UCD Memory Analysis Report Velocity Software
Monitor initialized:                      on 2066 serial 71CE3     First record
-------------------------------------------------------------------------------
Node/    <-------------------------Storage Sizes (in MegaBytes)------------->
Time/    <--Real Storage--> <-----SWAP Storage----> Total <--Storage in Use->
Date     Total  Avail Used  Total Avail Used  MIN   Avail Shared Buffer Cache
-------- ------ ----- ----- ----- ----- ----- ----- ----- ------ ------ -----
LINUXA    186.0   2.6 183.4 484.1 330.9 153.1  15.6 333.5      0    5.2 134.0
-------------------------------------------------------------------------------
13:49:00
LINUXA    186.0   2.5 183.5 484.1 324.8 159.2  15.6 327.3      0    2.5 137.7
-------------------------------------------------------------------------------
13:50:00
LINUXA    186.0   2.7 183.3 484.1 318.9 165.2  15.6 321.6      0    2.6 139.4
-------------------------------------------------------------------------------
13:51:00
LINUXA    186.0   1.6 184.4 484.1 306.9 177.1  15.6 308.6      0    3.3 139.0
-------------------------------------------------------------------------------
13:52:00
LINUXA    186.0   2.6 183.4 484.1 307.2 176.9  15.6 309.8      0    3.4 138.7
```

*Figure 11-17 ESAUCD2 Linux Memory Analysis 196MB run*

One more test was run, with storage reduced to 128 MB. That turned out to be too small to support this workload.

### 11.9.7 Network adapter monitoring

Some planning should be done for the OSA Adapter. In our test system, each Domino server had an OSA Adapter port. LinuxC had two ports, one for each Domino server supported by LinuxC. The aggregate rate across each adapter was between 70 K and 100 K bytes for this interval. Thus, a 1 Mbit adapter would not be suitable for this workload, and we were fortunate that these were 10 Mbit.

```
Report: ESATCP4  TCPIP Hardware Layer/Interfaces Report  Domino Redbook  ESAMAP 3.4.0 08/25/03 Page 70
Monitor initialized:   on 2066 serial 71CE3     First record analyzed: 08/24/03 18:00:00
-------------------------------------------------------------------------------------------
Date/        <Total Octets> Avg  <-Subnet packets / Sec-> <-----Packets Discarded------>
Time         <-Per second->   Q  <-Unicast-> <NonUnicast> <In Error> <NonError> Unknown
Node     IFT Input  Output  Len  Input Outpt Input Output Inpt Outpt Inpt Outpt Protocol
-------- --- -----  ------  ---- ----- ----- ----- ------ ---- ----- ---- ----- --------
08/24/03

19:00:00
MLOCAL   - 1  1502  485.32     0  2.69   2.0     0      0    0     0    0     0        0
LINUXA   - 1  1.67    1.67     0  0.03   0.0     0      0    0     0    0     0        0
LINUXA   - 3 46869   31819     0 55.39  50.2     0      0    0     0    0     0        0
LINUXC   - 1  6006  6006.2     0  4.59   4.6     0      0    0     0    0     0        0
LINUXC   - 3 43956       0     0 52.32     0     0      0    0     0    0     0        0
LINUXC   - 4 42773   64306     0 51.98 101.9     0      0    0     0    0     0        0
LINUXB   - 1  1.67    1.67     0  0.03   0.0     0      0    0     0    0     0        0
LINUXB   - 3 46883   33037     0 56.54  54.5     0      0    0     0    0     0        0
```

*Figure 11-18 ESATCP4 TCPIP hardware interface rate example for three-server run*

## 11.9.8 Disk/DASD

When moving existing workloads to other servers, the disk requirement is a known factor. In the following example, root requires 1507 megabytes of space. If this is shared between many servers, then this space is only required one time. Other volumes are more volatile, and disk space utilization of greater than 60 or 70% (such as in /domserva/notdesd at 94% space utilization) may be a risk. Thus, this data provides two items of importance in capacity planning for disk space: the information needed to transport workloads, and a value to check on a regular basis to avoid outages from disk full conditions.

```
Report: ESAHST2   LINUX HOST Storage Analysis Report   Domino Redbook  ESAMAP 3.4.0 08/25/03  Page 135
Monitor initialized:  on 2066 serial 71CE3   First record analyzed: 08/24/03 18:00:00
-----------------------------------------------------------------------------------------------------
NODE/           <-Utilization->          <------Storage------>
Time/           <MegaByte>  Pct          Alloc
Date     Index  Size  Used Full  Errors  Units  Description
-------- -----  ----  ---- ----  ------  -----  ------------------------
19:00:00
LINUXA
            1  2310  1507 65.2       0   4096  /
            2  2346  1877 80.0       0   4096  /opt
            3  2303   435 18.9       0   4096  /domserva
            4   16K  2872 17.8       0   4096  /notesmail/nbmai
            5   16K  2869 17.8       0   4096  /notesmail/nbmai
            6  2303  2178 94.6       0   4096  /domserva/notesd
          101   186   183 98.6       0   1024  Real Memory
          102   484  91.0 18.8       0   1024  Swap Space
          103     0     0    0       0    256  Memory Buffers
```

*Figure 11-19 ESAHST2 Linux host disk storage analysis example*

## 11.10 Special topics—z/VM storage planning

The difficulty in planning for storage (memory) when moving a workload from Intel or another environment is that “common knowledge” from those platforms does not apply well to Linux on z/VM environments. Recommendations for configuring for performance from these other platforms will rightfully recommend large amounts of storage to avoid swapping. This results in very large storage sizes.

### 11.10.1 Storage hierarchy

VM memory is usually called “storage”. Sometimes it is called “central procesor storage” or “real storage”. It can be called “RAM” by those using Linux terms. All virtual machines, virtual disks, minidisk cache (MDC), and other address spaces exist in VM storage. When storage in a virtual machine is not referenced for a period of time, there is no reason to maintain that storage in VM storage. This storage can be paged out by z/VM when memory is overcommitted and there is a need for other pages to be resident. Paging is either to DASD or to expanded storage (a section of processor memory).

One benefit of z/VM is the ability to define memory size at the level that each virtual machine requires. Some physical environments limit this to increments of 256 MB. So if 300 MB would do under VM, you would have to define 512 MB in some non-VM environments.

There is a storage hierarchy that z/VM has implemented for performance reasons. When there is expanded storage, z/VM will steal pages from virtual machines, virtual disks, and other address spaces and move those pages to expanded storage. When that becomes full and even more storage is required, z/VM will then migrate pages from expanded storage to the paging devices.

This three-level storage hierarchy is very efficient and provides a very effective caching architecture for the paging devices. In addition, these paging devices may have more levels of cache inside the storage processor. See Figure 11-20 on page 305 for an overall view of this hierarchy.

All virtual machine pages and virtual disk pages are first loaded into main storage. When these pages become idle, they become candidates for stealing and then for page migration, depending on current storage requirements. So the pages in main storage are meant to be the actively used pages—the pages currently in use. Because not all of the pages are active pages, and because some servers or virtual disks are idle at times, those pages can be paged out to make room for more work.

The term “overcommittment of storage” is sometimes used to describe how much storage has been defined as compared to how much storage is available. There is no good rule of thumb for the amount of over-commitment that provides good performance. As Linux and Linux applications become better players in this new shared storage configuration, the level of overcommittment that can be supported goes up. And there is a very large dependency on how many servers do go idle for periods of time and how long those periods of time are when the server does not need main storage.

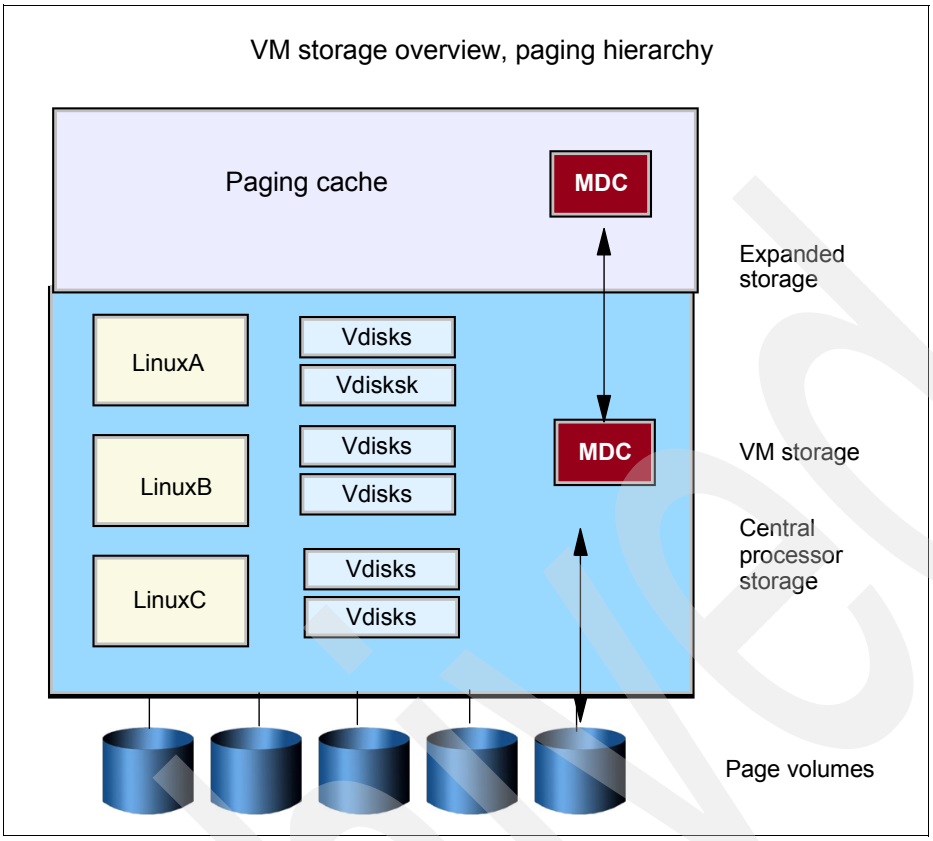



*Figure 11-20 Storage hierarchy*

There is also a concept of “working set” as compared to required pages. The working set refers to the active pages used by a server. Some of the pages might be used for initialization only or for error recovery. There is no need to leave pages in main storage that are never referenced.

Overcommitted is good; it is the only way to enforce sharing of resources. The paging algorithms in VM have been developed over many years, and very high page rates can be supported. However, you should not overcommit resources for a single Linux guest. The sum of virtual machine size and virtual disk space should not exceed what is available to z/VM.

## 11.10.2 Using virtual disk for swap

The standard Linux installation documentation often refers to a swap disk and places that device on real disk. It is important to understand why this can lead to poor performance if your storage requirements are larger than your virtual memory. As Linux starts using storage, and fills up RAM, it starts to move pages to the swap device. In times of heavy swapping, Linux could be attempting to swap hundreds of I/Os to a single device. This turns out to be a significant bottleneck. The only real caching hierarchy for this configuration is in the storage processor (DASD controller). See Figure 11-21 on page 306.

If there were 10 Linux servers and each of those servers had a real disk assigned as a swap device, there would now be 10 potential disks for serious performance problems. The load across these volumes would not be balanced, but would depend on each Linux server’s current requirements.

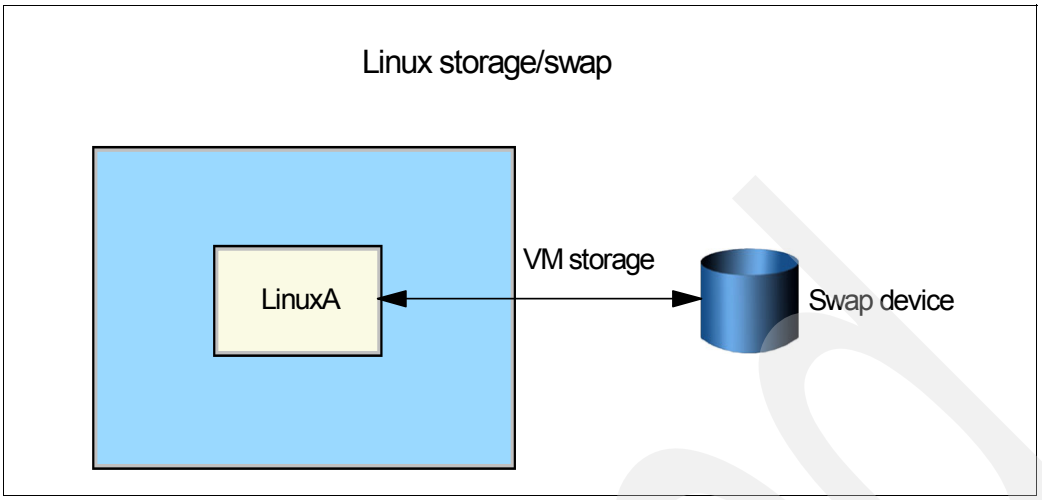



*Figure 11-21 Linux swap to real disk*

If those same 10 disks are instead turned into z/VM paging devices, and Linux uses VM virtual disk for swapping, there is now a three-level caching hierarchy that is backed up by 10 disks over which z/VM will balance any paging activity. Figure 11-22 shows a simplified illustration of this concept. In actuality, paging is not done through expanded storage, but through VM main storage.

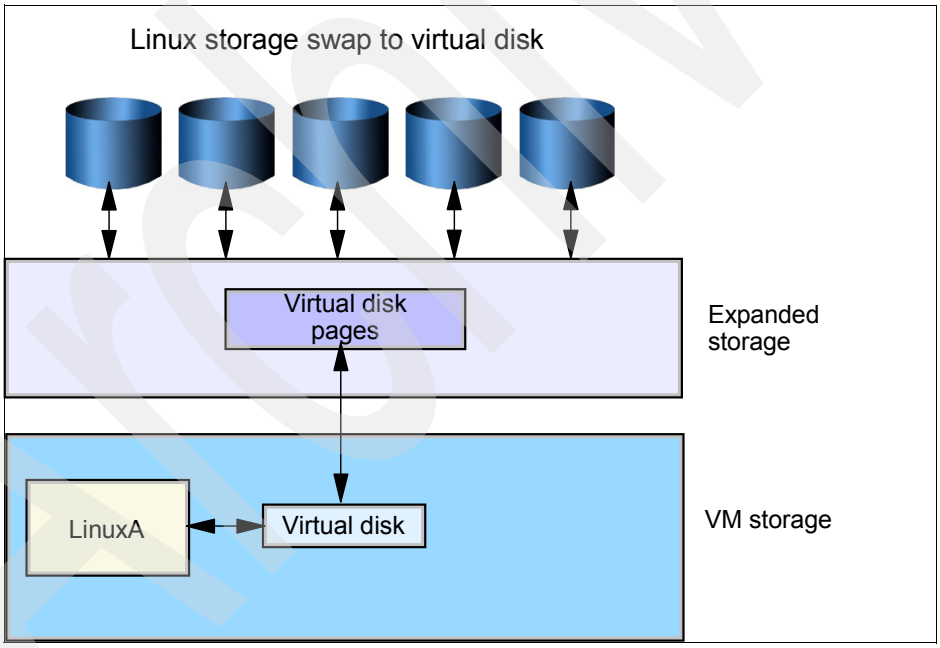



*Figure 11-22 Linux swap to virtual disk*

There are several advantages to using virtual disk. Most Linux and UNIX administrators are taught that swapping is bad and greatly impacts performance. But in the virtual disk case, running many tests with swapping at 1000 swap pages per second, very good performance was maintained. The reason for the difference is that virtual disks provide a storage hierarchy that has not been used before. Virtual disks really do avoid performance issues associated with swapping. However, there is a trade-off of increased CPU usage; see 11.10.3, “Virtual storage versus virtual disk swapping trade-off” on page 307.

Another benefit of using virtual disk is that the storage is not allocated until referenced by the Linux server. The cost of having 10 virtual disks for swap of 100 MB each is very small. The overall storage/disk requirements are greatly reduced—more swap can be defined than needed, and it will magically appear as an allocated resource only when called for.

### 11.10.3 Virtual storage versus virtual disk swapping trade-off

As virtual storage and overall storage requirements were dropped, CPU requirements to support swapping to the virtual disk had a cost. In the following measurement of the 256 MB virtual machine experiment with 1500 user load, swapping to virtual disk was about 1000 swap pages per second. The number of virtual disk I/Os is the measurement for swapping to virtual disk. For the four 15-minute intervals reported, there was an average of 850 K virtual disk I/Os, or almost 1000 swap pages per second over the 900 second reporting interval.

```
Report: ESAUSR3      User Resource Utilization - Part 2 Domino Redbook ESAMAP 3.4.0 08/25/03  Page 217
 Monitor initialized: on 2066 serial 71CE3 First record analyzed: 08/21/03 12:00:00
-----------------------------------------------------------------------------------------------
                 DASD MDisk Virt Cache I/O    <---Virtual Device---->
 UserID    DASD Block Cache Disk   Hit Prty   <----I/O Requests----->
 /Class     I/O   I/O  Hits  I/O   Pct Queued  Cons   U/R  CTCA Other
 -------- ----- ----- ----- ---- ------ ----- ----- ----- ----- -----
 08/21/03
 12:15:00  613K     0  248K 838K  74.8      0  1510     0   321     0
***Top User Analysis***
 LINUXA    610K     0  246K 838K  74.8      0     1     0     0     0
-----------------------------------------------------------------------
 12:30:00  615K     0  250K 822K  74.6      0  1487     0   324     0
***Top User Analysis***
 LINUXA    613K     0  248K 822K  74.6      0     0     0     0     0
-----------------------------------------------------------------------
 12:45:00  631K     0  260K 884K  75.5      0  1634     0   321     1
***Top User Analysis***
 LINUXA    628K     0  258K 884K  75.5      0     0     0     0     0
```

*Figure 11-23 ESAUSR3 Report showing virtual disk I/O for VM guest for 196 MB run*

The cost of performing this activity is charged to the kswapd daemon. In looking at the Host Application report in Figure 11-24 on page 309, the cost is about 9% of one of our processors to perform 1000 swap I/Os per second.

**Note:** This report shows the Linux guest CPU usage. The z/VM CPU usage must also be monitored in order to get the total picture.

#### Trade-offs to consider

When using virtual disk, monitor your system and consider these trade-offs; see 12.5 for additional discussion on this subject.

- Prior to z/VM 4.4.0, when a page fault occurs on a page of a virtual disk in storage, that page (and any other associated page for the fault) is brought into real memory below the 2 B address. This may add to the contention for storage below 2 GB on those systems. The page fault behavior was changed in z/VM 4.4.0,so that the page now is brought in above 2 GB where possible (as is done with general virtual server pages).
- While the pages backing the virtual disk blocks themselves are not created until referenced, the architected control blocks for dynamic address translation (page tables, segment tables, and so on) are created at the time the virtual disk is created. Further,

these control blocks are not pageable and must reside below the 2 GB address in real memory. This is another factor that can increase contention for storage below 2 GB.

- The VM storage management steal processing has a hierarchy of pages that it uses in trying to select the most appropriate pages. In that hierarchy, normal idle guest pages are lower on the hierarchy than a virtual disk in storage page. (A virtual disk in storage page is really a system utility space and therefore is given preferential treatment.) Linux guests often appear to stay in the dispatch list, never becoming dormant. A side effect of this is that guest pages are given close to equal priority with virtual disk in storage pages.

  However, over time as this is addressed and guests start to drop from the dispatch list, an undesirable effect may occur: Linux will determine an unused page and move it out to swap disk (a virtual disk in storage). If this guest goes idle, VM storage management will steal more aggressively from the guest pages (pages Linux decided it needed) and less aggressively from the virtual disk in storage (pages Linux decided it was safe to page out).

  The optimal case is when your Linux server has so much memory that you hardly ever swap. This means you do not waste response time and CPU cycles on swapping. Unfortunately that does not scale well, because in real life you cannot afford to give z/VM so much memory that you can line up all these big Linux virtual machines. Therefore, z/VM needs to page out portions of these Linux machines. That is unpleasant because:

  – A page fault on the first level may prevent Linux from doing anything useful (despite asynchronous page fault handling).

  – Linux and z/VM both implement a type of LRU algorithm to allocate pages, and these do not combine well.

  In conclusion, the sum of virtual machine size and Linux swap space depends on the application requirements. The virtual machine size depends on what you can afford (or are willing to give to) the Linux server. You cannot choose an optimal size without knowing the resources available on z/VM. Some tuning philosophies would just add more memory, but the complex interaction of Linux and z/VM could at times cause a high-cache, high-memory situation to result in more paging than if you limited each system to what it needed.

```
Report: ESAHSTA   LINUX HOST Application Report    Domino Redbook  ESAMAP 3.4.0 08/25/03  Page 298
Monitor initialized:    on 2066 serial 71CE3 record analyzed: 08/21/03 12:00:00
--------------------------------------------------------------------------------
 Node/    Process/    <-Application Process Counts------> <-----Processor---->
 Date     Application                                     <---Utilization---->
 Time     name        Total active Running ResWait Loaded Percent seconds Avg
 -------- ----------- ----- ------ ------- ------- ------ ------- ------- ----
 08/21/03
 12:15:00
 LINUXA   java         15.0   15.0     2.0    13.0      0    10.3    92.6  0.7
          kswapd        1.0    1.0       0     1.0      0     9.1    82.2  9.1  <------
          router       11.0   11.0       0    11.0      0    10.6    95.4  1.0
          server       67.0   67.0     1.0    63.0    3.0    63.2   568.5  0.9
          snmpd         1.0    1.0     1.0       0      0     3.3    29.3  3.3
          update        3.0    3.0     1.0     2.0      0    10.2    91.7  3.4
--------------------------------------------------------------------------------
 12:30:00
 LINUXA   java         17.0   17.0     2.0    15.0      0     9.5    85.9  0.6
          kswapd        1.0    1.0       0     1.0      0     8.8    79.5  8.8  <------
          router       12.0   12.0     2.0     9.0    1.0    11.0    99.3  0.9
          server       61.0   61.0     4.0    55.0    2.0    62.7   563.9  1.0
          snmpd         1.0    1.0     1.0       0      0     3.2    28.8  3.2
          update        4.0    4.0       0     4.0      0    12.0   107.8  3.0
--------------------------------------------------------------------------------
 12:45:00
 LINUXA   java         16.0   16.0       0    16.0      0    10.3    92.4  0.6
          kswapd        1.0    1.0       0     1.0      0     9.5    85.6  9.5  <------
          router       10.0   10.0       0    10.0      0    11.1    99.6  1.1
          server       67.0   67.0     9.0    53.0    5.0    64.3   578.6  1.0
          snmpd         1.0    1.0     1.0       0      0     2.4    21.9  2.4
          update        5.0    5.0       0     5.0      0    13.0   116.9  2.6
```

*Figure 11-24 ESAHSTA report for 196 MB run showing kswapd CPU requirement*

Our follow-on experiments were to reduce the machine size first to 196 MB, then to 128 MB. The run with 196 MB provided equivalent response time, with the swap rate averaging about 10% higher, and the kswap daemon also about 10% higher.

The 128 MB experiment proved the case that the swap rate is linearly proportional to the CPU required by the kswap daemon. At the one complete interval shown, swapping was 4 million for the interval, or over 4000 per second. The processor utilization was close to 200% across the two processors, as compared to about 130% across two processors for the previous run; see Figures Figure 11-25 on page 310 and Figure 11-26 on page 310.

```
Report: ESAUSR3  User Resource Utilization - Part 2    Domino Redbook   ESAMAP 3.4.0 08/25/03  Page 222
 Monitor initialized:     on 2066 serial 71CE3     First record analyzed: 08/21/03 15:30:00
 ---------------------------------------------------------------------------------------------------
                 DASD MDisk Virt Cache I/O    <---Virtual Device---->
 UserID    DASD Block Cache Disk   Hit Prty   <----I/O Requests----->
 /Class     I/O   I/O  Hits  I/O   Pct Queued  Cons   U/R  CTCA Other
 -------- ----- ----- ----- ---- ------ ----- ----- ----- ----- -----
 08/21/03
 15:45:00  220K     0  140K 133K  77.5      0  1039     0   322     1
***Top User Analysis***
 LINUXA    215K     0  136K 133K  77.4      0     0     0     0     0
-----------------------------------------------------------------------
 16:00:00  825K     0  581K   4M  94.5      0  1139     0   323     0
***Top User Analysis***
 LINUXA    817K     0  574K   4M  94.5      0     2     0     0     0
-----------------------------------------------------------------------
 16:15:00  354K     0  257K   2M  95.9      0  1037 13525   321     0
***Top User Analysis***
 LINUXA    351K     0  256K   2M  95.9      0     0     0     0     0
```

*Figure 11-25 ESAUSR3 showing virtual disk I/O 128 MB run*

```
Report: ESAHST1  LINUX HOST Software Analysis Report   Domino Redbook   ESAMAP 3.4.0 08/25/03  Page 277
 Monitor initialized:      on 2066 serial 71CE3    First record analyzed: 08/21/03 15:30:00
 ---------------------------------------------------------------------------
 Node/    <-----Software Program-------> <CPU Seconds> CPU  StgSize
 Time     Name      ID    Type   Status  Total Intrval Pct  (Bytes)
 -------- --------  ----  ------ ------- ----- ------- ---- -------
 08/21/03
15:45:00
 LINUXA   kswapd       8  Applic ResWait  20.4    18.4  2.0       0
-----------------------------------------------------------------------
 16:00:00
 LINUXA   kswapd       8  Applic ResWait 208.9   181.6 20.1       0
-----------------------------------------------------------------------
 16:15:00
 LINUXA   kswapd       8  Applic ResWait 273.7    63.7 10.7       0
```

*Figure 11-26 ESAHST1 process analysis showing kswapd CPU for 128 MB run*

If your virtual machine size is too small, you will begin to swap at a high rate. To reduce swapping, you can either allocate a larger virtual machine size or allocate virtual disks as your swap devices. Too much swapping increases CPU (charged to the kswap daemon), but too large of a virtual machine will increase the overall real storage requirement.

This will be a constant area for analysis and planning to utilize your current resources most effectively. When storage is an issue, review the size of the cache. If the cache size is excessive, then reducing the virtual machine size will reduce the size of the cache, providing some storage back for performing other work.

## 11.10.4 Minidisk cache (MDC) requirements

Another trade-off is the amount of storage allocated to minidisk cache, the benefit received, and the amount of storage left for the virtual storage of the virtual Linux servers. The benefits increase as more of the referenced data is shared across multiple Linux guests.

In the report shown in Figure 11-27 from the run supporting three servers, the minidisk cache hit ratio was 47 to 48% of all read I/O being performed by the servers. The relationship between the percentage of minidisk cache hits and the amount of storage allocated to minidisk cache is very workload-dependent. No conclusions should be drawn from this specific data, other than to understand the benefit being provided and the cost of that benefit. In this case, the minidisk cache was much larger that one would see in most environments. Under main store, the average size of the minidisk cache was 380 MB, and under expanded storage, an additional 2 GB was used. Typical production environments should see a total of less than 500 MB.

The size of the minidisk cache should be controlled, as z/VM will dynamically size the minidisk cache based on defaults that will not apply to these workloads. If paging becomes an issue, then reducing the size of the MDC from the default is normally a quick solution. The SET MDC command is used for this function.

Using the diagnose I/O driver will also impact the MDC storage requirement. When the volumes are formatted to allow the use of diagnose I/O, MDC can be tuned to use a record cache instead of a track cache. This is at the minidisk level. If MDC cache size requirement is too large, using the Diagnose driver with record cache is an option that should be evaluated.

```
Report: ESAMDC       Minidisk Cache Analysis       Domino Redbook      ESAMAP 3.4.0 08/25/03  Page 211
Monitor initialized:                 on 2066 serial 71CE3     First record analyzed: 08/24/03 18:00:00
-----------------------------------------------------------------------------------------------------
         <----Load---->     <IO per><Insertions> <-----Main Storage MDC--> <-Expanded Storage MDC------>
         <-Users-> Tran Hit <second> Usr Per Not <-Sizes (MB)--> </Second> <-Sizes (MB)--> <Per Second >
Time     Actv In Q /sec Pct rds hits Max Min Ald Avg MIN MAX Obj Stls Delt Avg MIN MAX Obj Rds Wrts Stls
-------- ---- ---- ---- --- --- ---- --- --- --- --- --- --- --- ---- ---- --- --- --- --- --- ---- ----
08/24/03
19:00:00   20  7.9  4.0  48 376  181 17K 6.4   0 380   0  2K  1K    0    0  2K   0  2K  2K  2K 47.0  4.7
20:00:00   20  8.1  4.0  47 370  176 17K 6.4   0 373   0  2K  1K    0    0  2K   0  2K  2K  2K 46.0  4.9
21:00:00   20  7.7  3.9  48 377  179 16K 6.6   0 375   0  2K  1K    0    0  2K   0  2K  2K  2K 46.5  5.0
**********************************Summary*****************************************************************
Average:   20  7.9  4.0  48 374  178 17K 6.5   0 376   0  2K  1K    0    0  2K   0  2K  2K  2K 46.5  4.9
```

*Figure 11-27 ESAMDC analysis for three-server run*

## 11.10.5 Operating system service levels

The technology for supporting Linux under z/VM is improving rapidly. From a Linux view, ensure you are taking advantage of the timer patch. The normal Linux timer pops every 10ms by default, or 100 times a second. The impact of this is twofold—one, it uses processor time and two, it keeps the virtual machine active and less eligible to page. The patch switches this timer pop to run only when necessary.

When you share resources with other virtual machines, it is not wise to install applications or daemons that you do not need. See 10.3 for more information about installing the timer patch. Also see *Linux on IBM @server zSeries and S/390: Performance Measurement and Tuning*, SG24-6926 for information how to analyze the timer ticks when the virtual machine does not drop from the queue when idle.

The next requirement is for a more recent Qdrop APAR that applies to z/VM 4.3 and z/VM 4.4. The number is VM63282. The impact of this APAR allows servers using qdio, HiperSockets, and vctca for communications to drop from queue. This allows those servers to be paged out and their page working sets to be calculated effectively. The impact of this APAR has not been analyzed, but expectations are that overall storage requirements will drop significantly. This APAR was installed as part of the project, but not evaluated to determine its impact.

## 11.11 Special topics—z/VM paging capacity planning

Page space has two requirements: sufficient space, and enough devices. The space calculation is very simple: add up all of the virtual machine sizes you will support, add up all of the virtual disks that will be used, and multiply by 4. This is conservative, but will ensure that you do not run out of storage.

The multiplier of 4 is for performance reasons. As available page space decreases, so will the block size, and then the I/O requirement will go up accordingly. In practice, if some amount of attention is paid, a multiplier of 2 will provide the same performance. But then if several Linux servers are added, you might be down to 1 or less before an opportunity to analyze the page space is taken.

The device calculation is based on workload and device technology. A target maximum device busy of 20% is reasonable. If you are paging to the RVA devices on ESCON channels (like those in Figure 12-26 on page 336) with 15 to 20ms response time, then each device can support about 10 I/Os per second.

However, if the page devices are on ESS with FICON channels with 1 to 2ms response time, then each device will support over 100 I/Os per second. Since we had no opportunity to run tests in a storage-constrained environment, there was very little paging.

### 11.11.1 z/VM page space versus Linux swap space

There is a trade-off between having a Linux swap disk on real versus virtual disk. When the swap space is on a real disk, it will be a single point of contention. The sustained swap rate can reach 100 or more, depending on your DASD. At this point, the Linux server will have very poor performance. When the Linux swap space is on virtual disk, there is no additional resource cost until the virtual disk is used.

If there were 50 servers, each with 2 GB of real disk allocated for swap, there is a 100 GB requirement. Since each of those disks are dedicated to a server, all swap activity from one server will go to one disk.

The other option is to define those 50 disks as page devices. Under times of very heavy swapping, and when z/VM does need to offload pages, the I/O is now spread over the 50 page devices. This increases the total real I/O bandwidth potential by each individual server to 50 times what could be supported on just one device. The total supportable load for the system has increased significantly, and provides the ability for each individual Linux guest to page at very high rates with no ill effect on performance. This is a tremendous advantage of virtual disk over any other swapping technology.

## 11.12 Linux swapping

Here are two performance tips for Linux swapping:

- Use the diagnose driver for swap, as discussed in “Make virtual disk space available to the z/VM system” on page 126.
- Use multiple prioritized virtual disks. If the overall requirement is estimated at 1 GB of swap space, then having 10 disks of 100 MB each will reduce the load on each individual disk. However, keep in mind the caveats listed in “Trade-offs to consider” on page 307. CPU is used during swapping, whether to one disk or to multiple disks.

Keep in mind that virtual disks are *not* allocated prior to use. Linux sees the virtual disk as a full disk ready for use. But until Linux writes something to this disk, z/VM will not allocate resources. Thus, the cost of many large disks is really very small until the disk is utilized.

Each of these swap devices should have a different priority. Linux allocates space on the highest priority disk in a moving cursor algorithm, which causes writes to be written serially across a disk until that disk is full. At that point, Linux will re-use available space across that disk until it is again full. Then Linux will use the next priority disk.

Linux will prioritize all of the virtual disks allocated to swapping, and completely use one before using the next. If there were just one big disk, that disk would be completely utilized. Using prioritized smaller disks reduces the load on your storage and paging subsystem because the Linux footprint is kept much smaller.

In Figure 11-20 on page 305, note that LinuxC has 10 virtual disks, but the total cost to the systems is 9 MB. This is pageable when there is a need for VM to page out idle pages. In the LinuxA virtual disks, the cost of 7 of the disks is only 75 pages, with the rest already paged out to disk.

```
Screen: ESAVDSK  ITSO z/Domino Redbook          ESAMON V3.3  08/26 18:08-18:09
2 of 3  VDISK Analysis                          SPACE * USER *       2066 71CE3

                                          <--pages-->  DASD    X-
                                          Resi- Lock-  Page Store
Time     Owner    Space Name              dent    ed  Slots  Blks
-------- -------- ------------------------ ----- ----- ----- -----
18:09:00 LINUXA   VDISK$LINUXA$$$021A$01AD      0     0   105     0
         LINUXA   VDISK$LINUXA$$$021B$01AE      0     0   105     0
         LINUXA   VDISK$LINUXA$$$021C$01AF      0     0   105     0
         LINUXA   VDISK$LINUXA$$$021D$01B0      0     0   105     0
         LINUXA   VDISK$LINUXA$$$0214$01A7  12235     0   649    13
         LINUXA   VDISK$LINUXA$$$0215$01A8  12369     0   223    15
         LINUXA   VDISK$LINUXA$$$0216$01A9  11843     0  1323   239
         LINUXA   VDISK$LINUXA$$$0217$01AA     75     0   105     0
         LINUXA   VDISK$LINUXA$$$0218$01AB      0     0   105     0
         LINUXA   VDISK$LINUXA$$$0219$01AC      0     0   105     0
         LINUXC   VDISK$LINUXC$$$0301$0201    208     0     0     0
         LINUXC   VDISK$LINUXC$$$0302$0202    208     0     0     0
         LINUXC   VDISK$LINUXC$$$0303$0203    208     0     0     0
         LINUXC   VDISK$LINUXC$$$0304$0204    208     0     0     0
         LINUXC   VDISK$LINUXC$$$0305$0205    208     0     0     0
         LINUXC   VDISK$LINUXC$$$0306$0206    208     0     0     0
         LINUXC   VDISK$LINUXC$$$0307$0207    208     0     0     0
         LINUXC   VDISK$LINUXC$$$0308$0208    208     0     0     0
         LINUXC   VDISK$LINUXC$$$0309$0209    208     0     0     0
         LINUXC   VDISK$LINUXC$$$0310$020A    208     0     0     0
```

*Figure 11-28 ESAVDSK virtual disk report showing swap disk requirements*

To reduce the footprint of Linux storage, we recommend that you give Linux some swap space in the VM virtual disk and reduce the virtual machine size by the same amount. Although the virtual machine primary address space and virtual disk come out of the same z/VM resource set (main storage and page space), the benefit is that Linux's usage of the two is different. Swapping to virtual disk is pretty fast, so the penalty for some occasional swapping is not too high. The effect is that the footprint of the Linux machine (the sum of resident primary address space and virtual disk) is reduced, which is desirable.

There is a myth that swap space should be twice the virtual machine size. But you can have too much of a good thing, and so to build virtual machines of 512 MB with 1 GB virtual disk can be a wrong decision. Initially you can get away with it, since Linux has not touched the virtual disk so it does not take any real storage yet. We find that even an oversized Linux virtual machine will do some swapping now and then. We recommend that you lower the virtual machine size to the point where you occasionally swap a little bit, so most systems do a bit more than that.

Because of the way pages are allocated on the swap disk, over time Linux will have touched most pages in the virtual disk. These pages contain no active data any more (swapping is normally read back only once) but we currently have no way to tell z/VM about that. So z/VM needs to back those virtual disk pages with pages from your paging subsystem. This results in a footprint of 1.5 GB for your virtual machine. When you have touched all virtual disk pages, things get worse because the reference pattern is such that each swap-out by Linux requires a page-in by z/VM.

See 11.10, “Special topics—z/VM storage planning” on page 304 and 12.7.7, “Swapping (Linux)” on page 338 for additional discussion and recommendations on swapping.

12

# Performance monitoring and tuning

After you have installed Domino, monitoring and managing your environment becomes critical to a successful implementation. You have a variety of options for running Domino, from native Linux to a guest under VM. Also, with Domino Release 7, platform statistics can be enabled. This provides other options with the Domino Administration Client.

In this chapter, we discuss native Domino options (including some Release 7 features) for performance monitoring and tuning that apply to native Linux zSeries LPARs as well as Linux guests under z/VM. We also discuss some options from z/VM, specifically the IBM performance toolset and ESALPS from Velocity Software. Note: This is not a complete list or a recommendation for any one of these toolsets.

## 12.1 Overview

This redbook provides guidance on implementing, measuring, and optimizing performance using a zSeries configuration. The workloads used were chosen to show the processes used in measuring and analyzing performance. Any measurements provided should not be construed as a benchmark.

No single tool can adequately monitor and tune Domino on Linux, whether in native LPARs or as guests under z/VM. System level tools that deal with the complexity of the zSeries do not deal with the complexity of Domino in a cross platform environment. Also, the Domino performance analysis features do not understand the complexity of the zSeries. Your success in managing your Domino environment will hinge on merging the various data sources. While either Domino or the operating system can detect an issue, understanding it and resolving it could require analysis using tools in the other area.

Be careful about using rules of thumb, because Domino provides a wide range of services and features. Depending on what you have enabled and how you manage it, you may find that two servers with the same number of users can show dramatically different resource usage. It is isn't that the server with more CPU is inefficient; rather it is how those users operate. It is common even within a single customer to have servers that consume that same amount of CPU but support a very different number of users. Many things (how active are your users, how big do you allow their mailboxes to grow, do you manage in-boxes, do you allow personal agents) have an effect on the amount of CPU required to support a user community. Managing your servers by resource utilization, not registered users, will help you avoid a number of potential pitfalls.

**Caution:** Linux on all platforms uses wall clock time, not dispatch time, in reporting CPU numbers. It is possible with shared engines for native Linux LPARS or guests under z/VM to report more CPU usage than the actual. While you can normalize Linux reports with z/VM or LPAR manager data to get the correct total usage, any data from within Linux would assume a even distribution of CPU in contention periods. The fact that you have these contention periods can be worthwhile to note; abnormal behavior during these periods would indicate something very different from abnormal behavior during a lack of contention.

**Note:** Domino's ability to report platform process statistics running under Linux on any platform is a new feature in Release 7.

## 12.2 Additional information

Additional information about the various products and data sources that are discussed in the this chapter can be found at the links below:

For additional information about performance measurement and tuning for running Linux as a z/VM guest on IBM @server zSeries and S/390 machines, refer to *Linux on IBM @server zSeries and S/390: Performance Measurement and Tuning*, SG24-6926, which was written by some of the same authors who wrote this redbook.

For additional information about the new platform process statistics in Domino for Linux, refer to the Internet article:

http://www-128.ibm.com/developerworks/lotus/library/article/stats-linux/

For additional information about Linux on IBM @server zSeries and S/390: Performance Toolkit for VM, refer to:

http://www.redbooks.ibm.com/abstracts/sq246059.html?Open

For additional information about the z/VM Performance Toolkit, refer to:

http://publibz.boulder.bim.com/epubs/pdf/hcsl7b00.pdf

For additional information about RMF/PM, refer to:

http://www-1.ibm.com/servers/eserver/zseries/zos/rmf/rmfhtmls/pmweb/pmlin.htm

For additional information about Domino Administration Client, refer to the Admin Client help that ships with the Notes Administration Client.

For additional information about SAR data, refer to:

http://perso.wanadoo.fr/sebastien.godard/

## 12.3 Domino Administration Client

With the advent of Domino Release 6, the administration client was enhanced to integrate the platform statistics that are being collected by Domino on the various platforms. Figure 12-1 shows a sample of the Administration client using real-time monitoring on a Domino Server. This server could be on any of the supported Domino platforms. This client provides the Notes Administrator with a good top-down view into the DPARs of what is running and how much CPU resource it is taking.

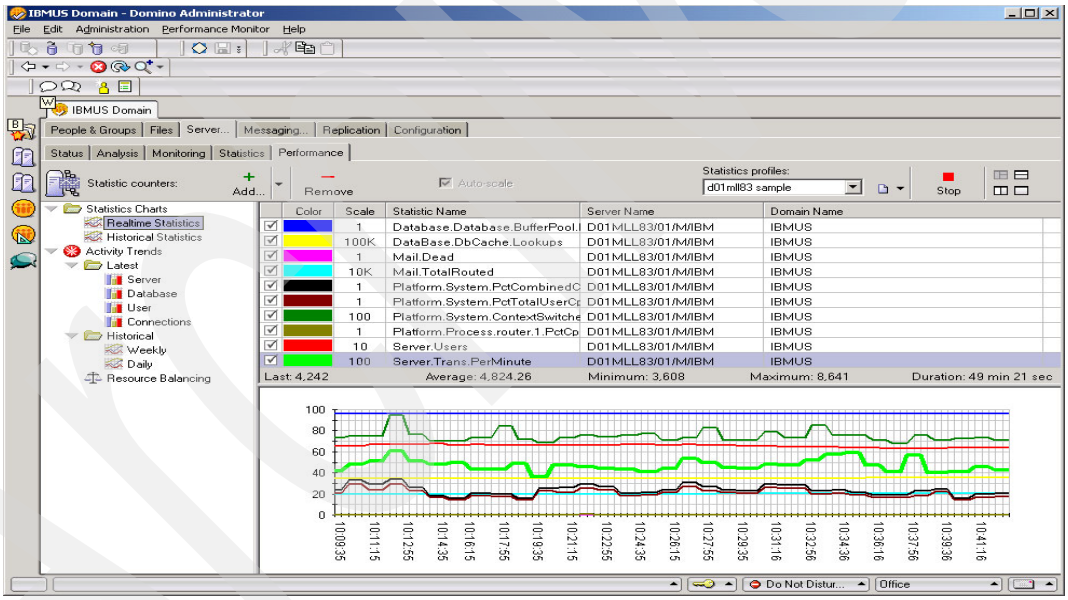



*Figure 12-1 Real-time monitoring of a server*

In addition to the real-time view, you can use the Admin client to see the historical views into the data. Figure 12-2 on page 318 shows the past trend of transaction counts for this Linux DPAR.

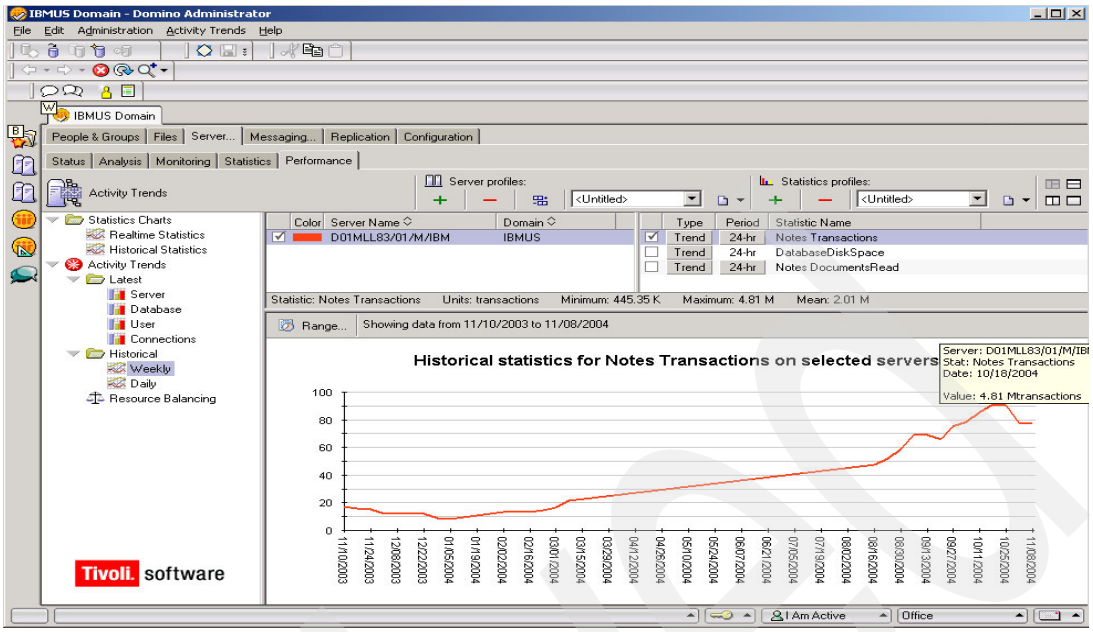



*Figure 12-2 Historical views*

In addition to the charting capabilities, the Admin Client give you real time access to the full range on Domino statistics, as Figure 12-3 shows.

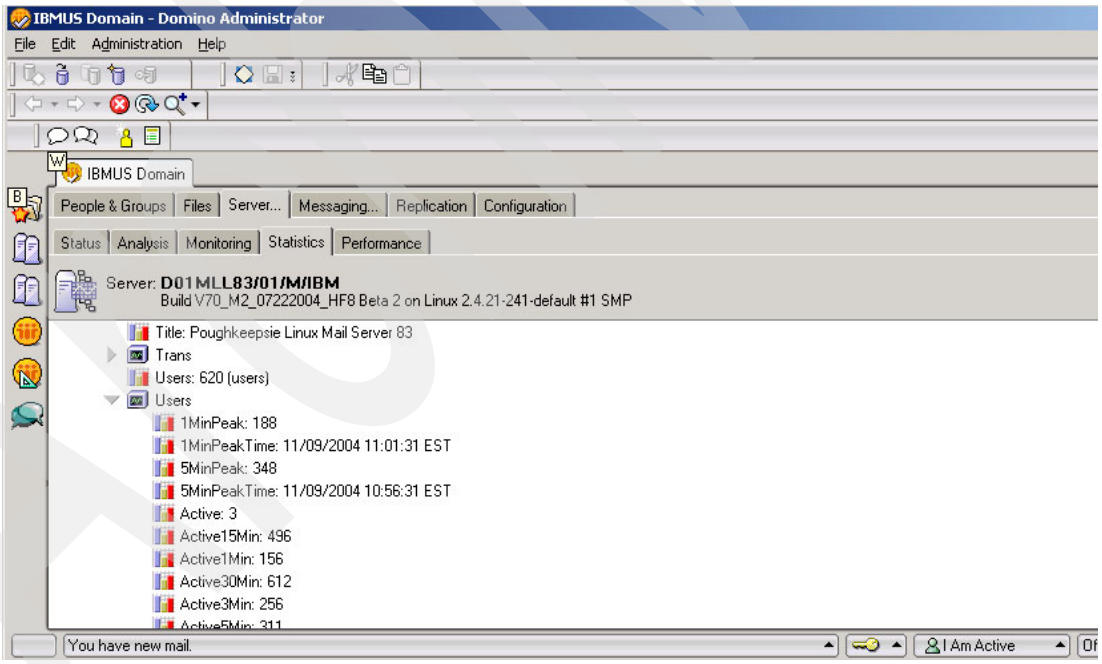



*Figure 12-3 Domino statistics*

By using a combination of charting and drilling down, you can get both the real time and the historical perspective on the various types of workloads you are asking your Domino servers to perform.

**Caution:** Domino typically stores its statistics as cumulative fields. These are the totals since the server was started or the value was explicitly reset. When looking at statrep historical

data, you will need a beginning and an ending data point to understand what actually happened during that interval.

The Server Health Monitor feature allows you to look at how well a server is doing overall. See Figure 12-4.

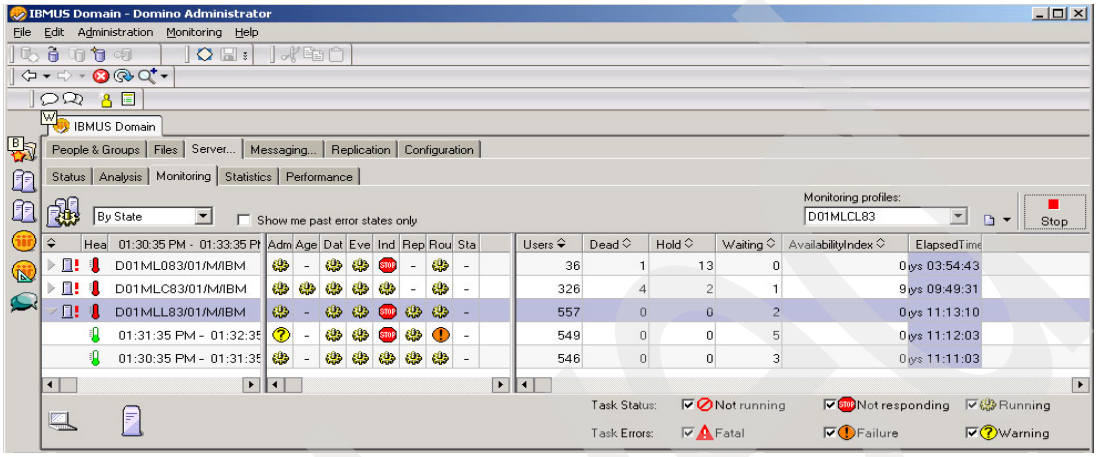



*Figure 12-4 Server Health Monitor*

Once you see a server that needs attention, you can drill down further to see additional details. See Figure 12-5.

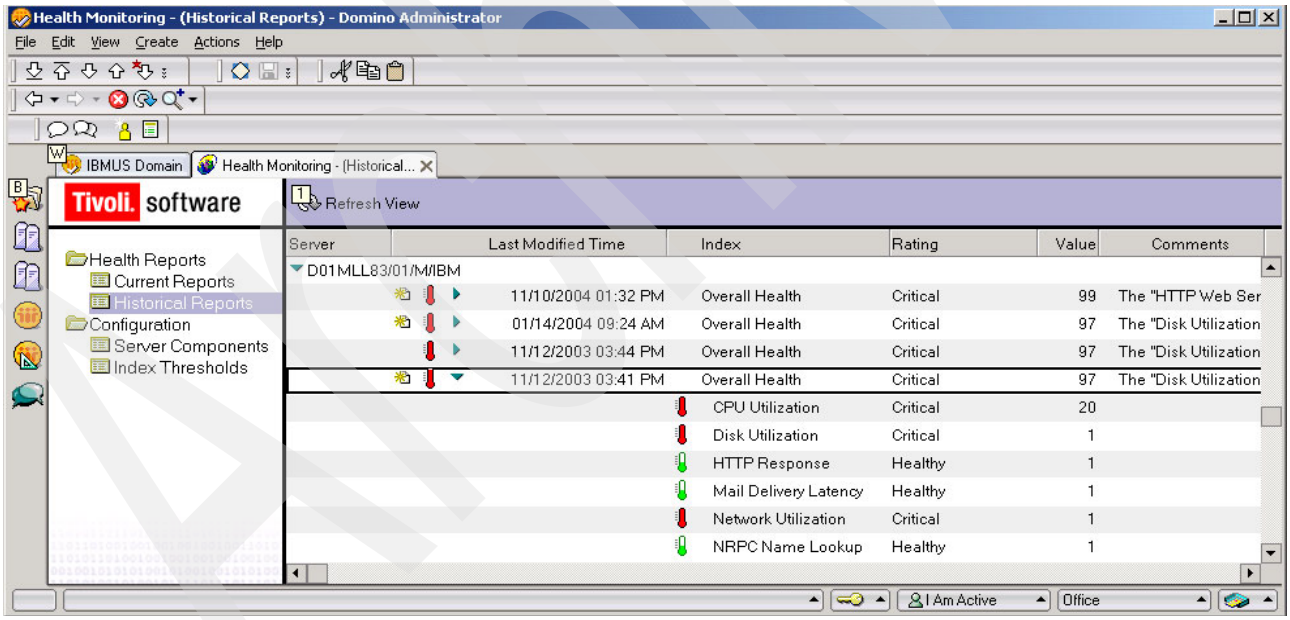



*Figure 12-5 Server health details*

While the Admin client can provide you with detailed analysis of your Domino servers, it will not provide you with details of the Linux OS or hardware. You will need to use other tools that understand LPARs, guest machines, and shared engines.

If a statistic is increasing in value, you need to find out when. Domino has a statistic called Database.DbCache.OvercrowdingRejections. This tells how many times that Domino opened a database and built its metadata, and then, because the DbCache was full, threw that information away. The next time that you access that database, you must reopen it since it will

not be in the DbCache. OvercrowdingRejections an have a negative impact on your performance. Figure 12-6 shows an export for a Domino Statrep database looking at the OvercrowdingRejectons value.‘

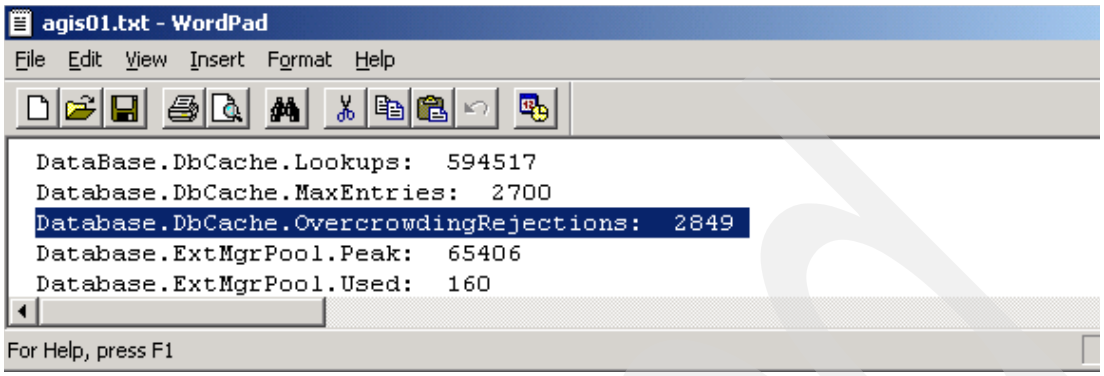



*Figure 12-6 Overcrowding rejections*

We can see that the server has had a number of database rejections. Is this good or bad? The answer is, it depends. Figure 12-7 shows a plotting of the DbCache Rejects over a three-week period of several servers. This chart show the interval values for this statistic. You can see that, at its peak, there were 18,000 rejects in a 15 minute interval.

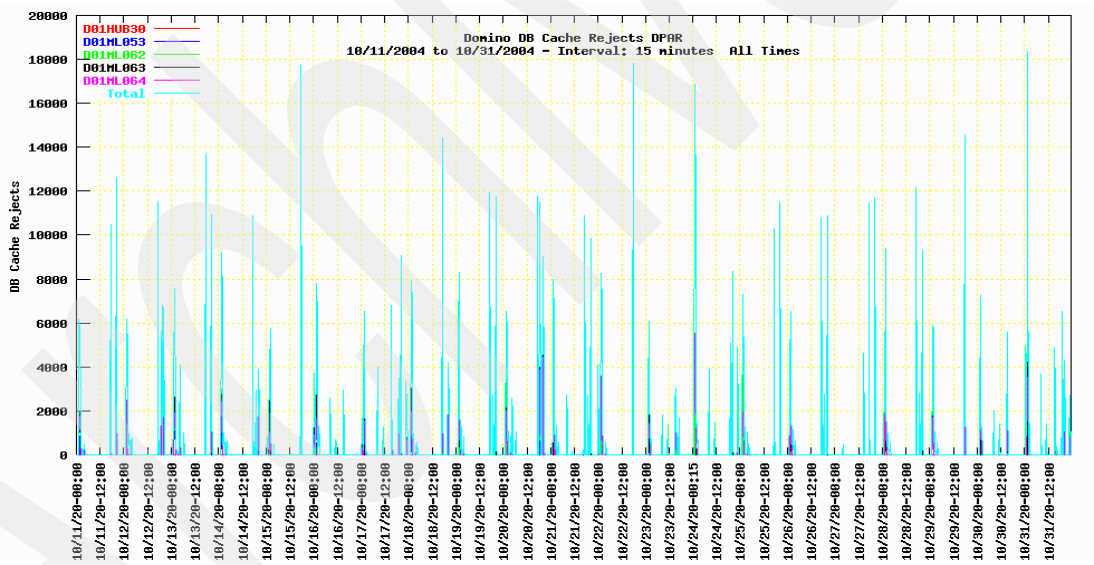



*Figure 12-7 DbCache rejects*

If we plot the same data for just prime shift (see Figure 12-8 on page 321), we see a very different picture.

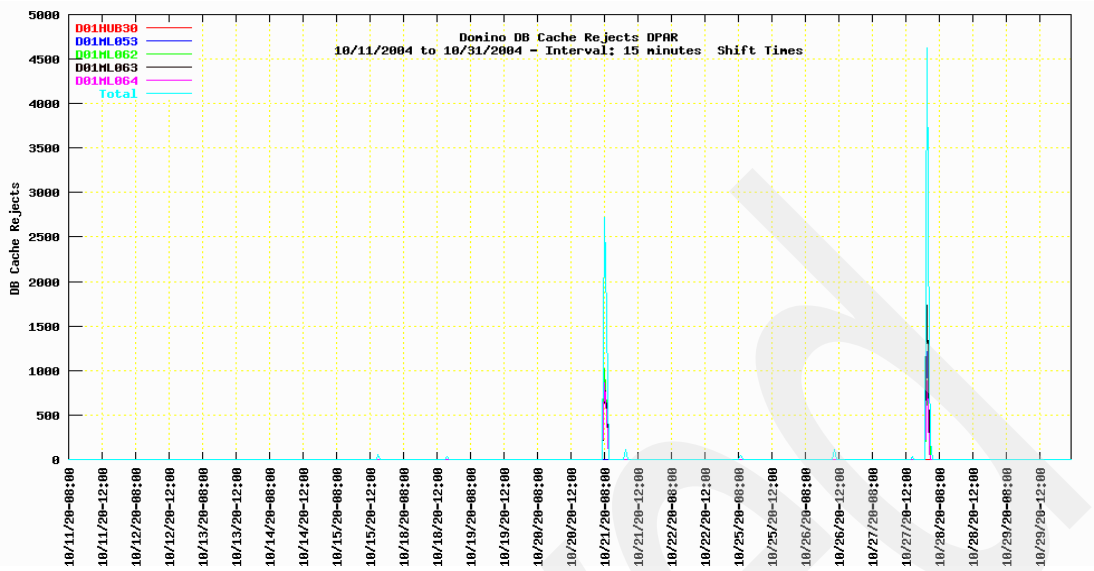



*Figure 12-8 DbCache rejects during prime shift*

Almost all of the rejects are occurring outside of the prime shift window. Since the ServerAt task will wake up during the night and access most of the databases sequentially, we expect the DbCache to be overrun. Since Domino does not re-access these databases during the run, there is little benefit to having them in the cache. So knowing when the counters are increasing is just as important as knowing that they are increasing in the first place.

## 12.4 Domino domain monitoring

In looking for future improvements to Domino, customers were asked to identify the most costly and difficult areas of Domino administration. Figure 12-9 on page 322 represents their feedback. It reveals that monitoring and troubleshooting servers and clients were the most costly areas. While many of the client issues were addressed with Release 6, the server-side issues were not. In Release 7 of Domino, a feature called Domino Domain Monitoring (DDM) was created to address the area of monitoring and troubleshooting servers.

## Administration

- At the highest level, reduces TCO by providing a means to quickly monitor and determine the health of an entire domain at a single UI location and quickly resolve problems.
- Automate problem determination/analysis by feature (replication, messaging, directory, security, etc)
- Rollup, prioritize and resolve problems across servers
- Hide details until they are needed
- Make DDM checks and reporting configurable and flexible
- Pave the way for further autonomic initiatives

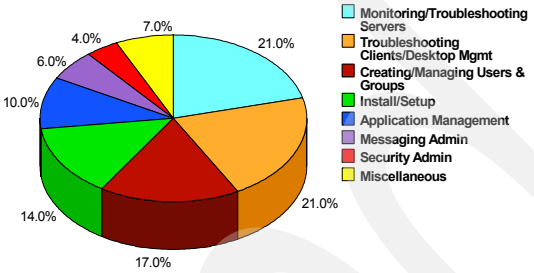



*Figure 12-9 Customer administration requests*

DDM will allow administrators to manage and monitor the health of multiple servers across multiple domains from a single control point. DDM brings a new administration paradigm to Domino troubleshooting. Its reporting interface rolls up, prioritizes, and consolidates errors and event notifications, preventing administrators from having to visit many different interfaces on different servers to discover problems or issues. If the same error or event occurs more than once, DDM collects those occurrences and presents them in a single event document. It provides the most crucial high level information first and allows drill-down to see additional details and complexity, preventing information overload.

A new database (DDM.NSF) serves as the reporting UI and data repository for DDM and non- or pre-DDM events. DDM configuration is simple and built into events4.nsf, where event task configuration already existed for pre-Domino 7.0 releases. DDM provides close to 100 highly configurable probes that actively check and instrument different server feature areas—directory, security, Web server, replication, agents—that report critical information back to DDM. Each server or group of servers can have different probes set to different threshold, schedule, and content-checking configuration levels. This allows administrators to monitor a hub server differently from a mail server or an application server in the same domain. It can handle enterprise diversity in Domino topologies.

Figure 12-10 on page 323 shows a beta example of high level information reported by DDM probes into the DDM database UI.

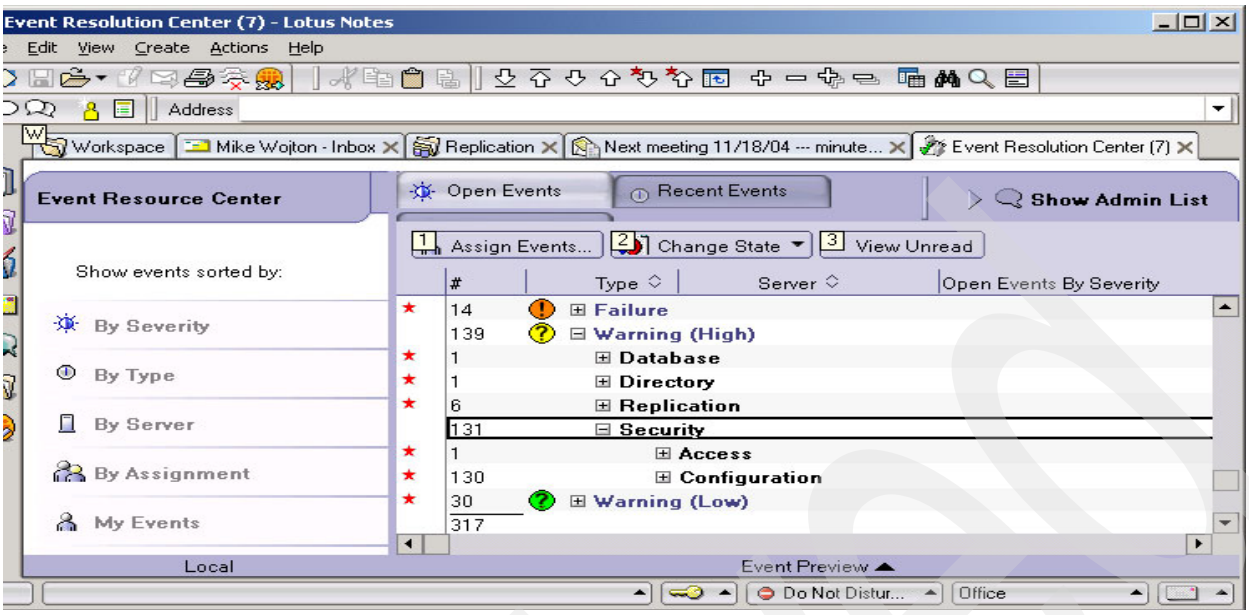



*Figure 12-10 DDM report*

The UI allows the most severe problems across the entire domain to be viewed first. Event documents can be opened and the administrator can drill down to the specific information for that issue; see Figure 12-11. In many cases, probable cause and possible solution information is provided, along with corrective actions that will take administrators to the right tool or server document to resolve the problem.

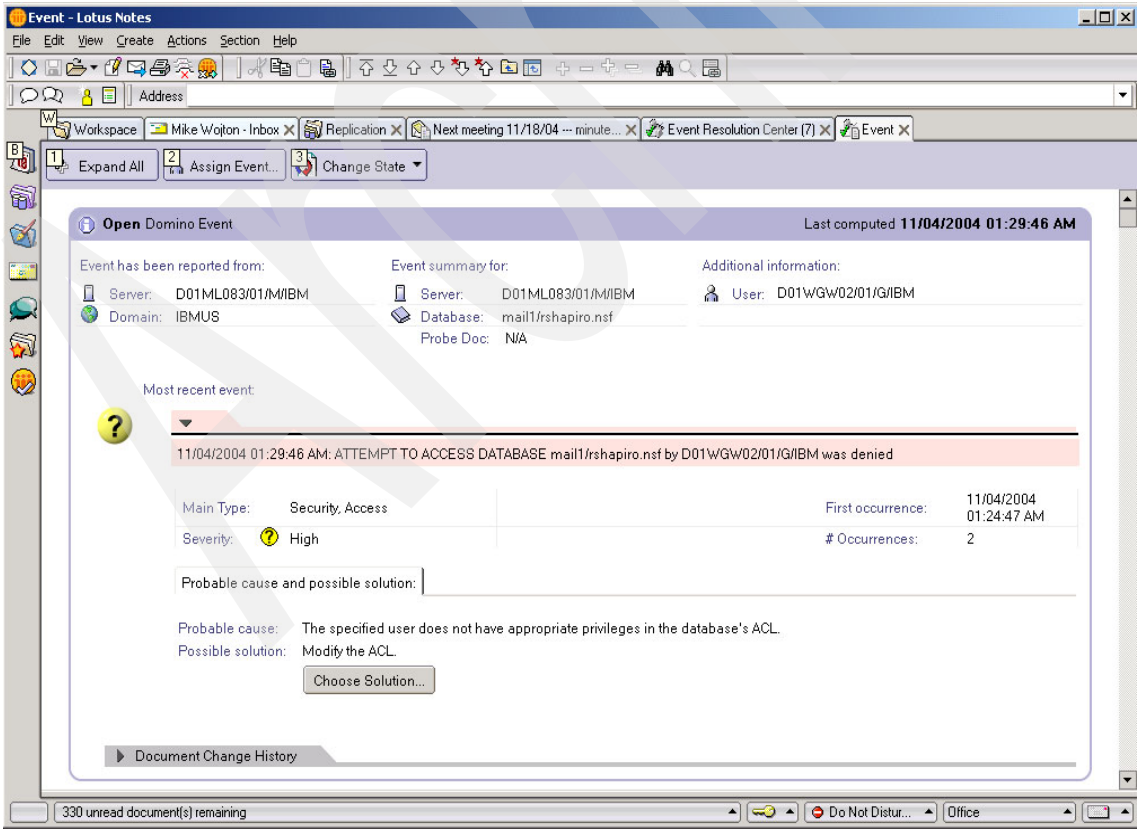



*Figure 12-11 DDM drill down*

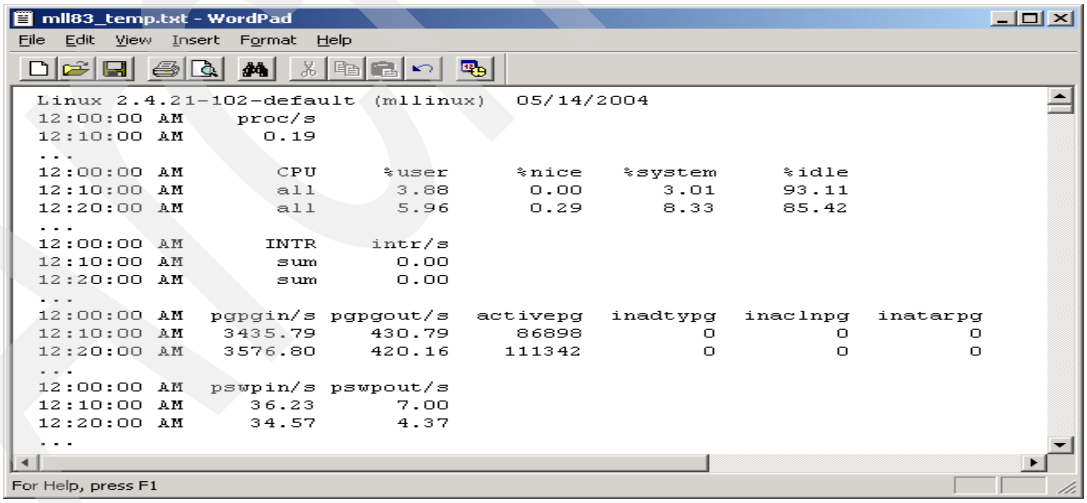
DDM collection hierarchies allow information on multiple servers to be reported to a single server or other servers in a multi-tiered hierarchy, where that event information is prioritized and consolidated. Collection hierarchies can be configured any the administrator desires: to represent administrative responsibilities or regions or organization structure. For example, all servers at a particular site could report to a single departmental server. That server and others could then feed all of that information up to regional servers, and then to one or more servers at a higher corporate level. Again, this provides administrators with a powerful event filtering and rollup tool, facilitating management of a great deal of Domino event information, and ultimately server health.

## 12.5 Linux SAR data

In addition the real time monitoring tools, Linux stores information about the performance of the OS into a SAR data set (???statistics and reporting???). This data can then be reported on with SAR function in the Linux OS. You can use this data for performance and capacity analysis on your systems at a system level. This data will not show you how the individual components of your Domino Server are running, but will give you an in-depth view in of your CPU, memory, and I/O usage.

If you are running native Linux LPARs, you do not have the VM monitoring tools, so you need to exploit the SAR data or use another tool such as RMF/PM. However, there will be more details in the SAR data than from the other tools, including VM. Understanding the data that is available to you from each tool will greatly assist you with performance issue by knowing what you have to drill down into when a problem arises.

The SAR command can be used to set you system collection and interval recording of your system's data. It is also used to post process the binary machine data file into readable reports. You can specify a wide range of reports or a single report in a SAR run from your data. Figure 12-12 shows a sample of some reports from a SAR run using the -A options




```
Linux 2.4.21-102-default (mllinux)   05/14/2004
12:00:00 AM     proc/s
12:10:00 AM       0.19
...
12:00:00 AM       CPU      %user      %nice    %system      %idle
12:10:00 AM       all       3.88       0.00       3.01      93.11
12:20:00 AM       all       5.96       0.29       8.33      85.42
...
12:00:00 AM      INTR     intr/s
12:10:00 AM       sum       0.00
12:20:00 AM       sum       0.00
...
12:00:00 AM  pgpgin/s pgpgout/s  activepg  inadtypg  inaclnpg  inatarpg
12:10:00 AM   3435.79    430.79     86898         0         0         0
12:20:00 AM   3576.80    420.16    111342         0         0         0
...
12:00:00 AM  pswpin/s pswpout/s
12:10:00 AM     36.23      7.00
12:20:00 AM     34.57      4.37
...
```

*Figure 12-12 SAR reports*

The data is this raw format can provide information about your system's resource utilization. However, the data in this format is not very usable for multiple days/weeks of analysis. This data can be loaded into a spreadsheet or a relational database for reporting and graphing. The chart below shows the prime shift (8 AM to 4 PM) CPU usage for about four weeks in a Linux LPAR.

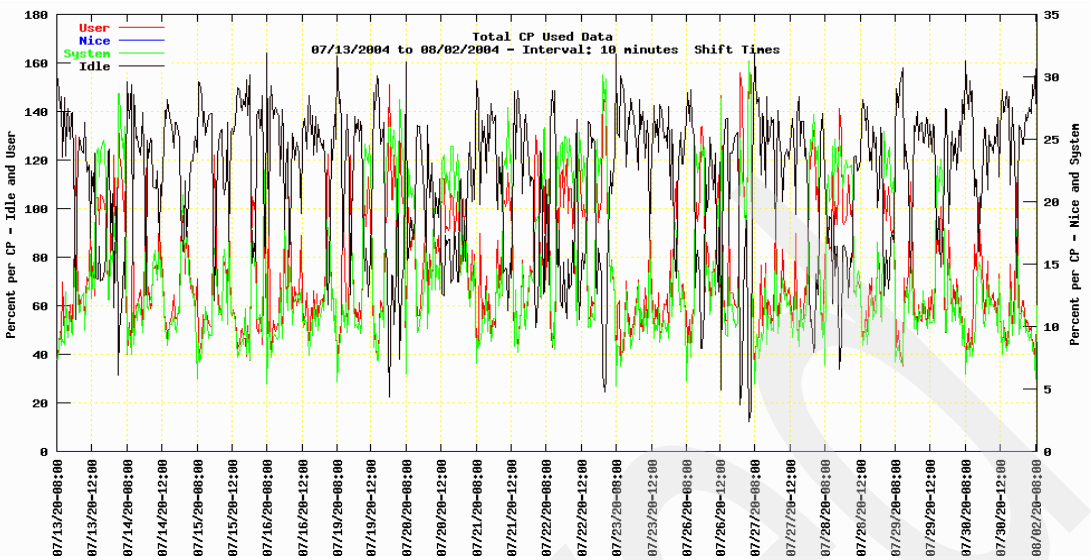



*Figure 12-13 Total CPU usage*

You will notice that the CPU is over 100%. This is because SAR reports usage based on a single CP. This LPAR has two engines available to it so the total across the fields will be 200%. Notice that there are two Y axis scales or the above chart. This is to handle the different numbers. By plotting to two scales, we can see greater details in the data that has a much lower values. Fluctuations in these lines would be squashed and lost if we were to plot them all on a single Y axis.

Plotting the same data on a daily basis, again from prime shift only, provides a capacity trend for this LPAR. See Figure 12-14.

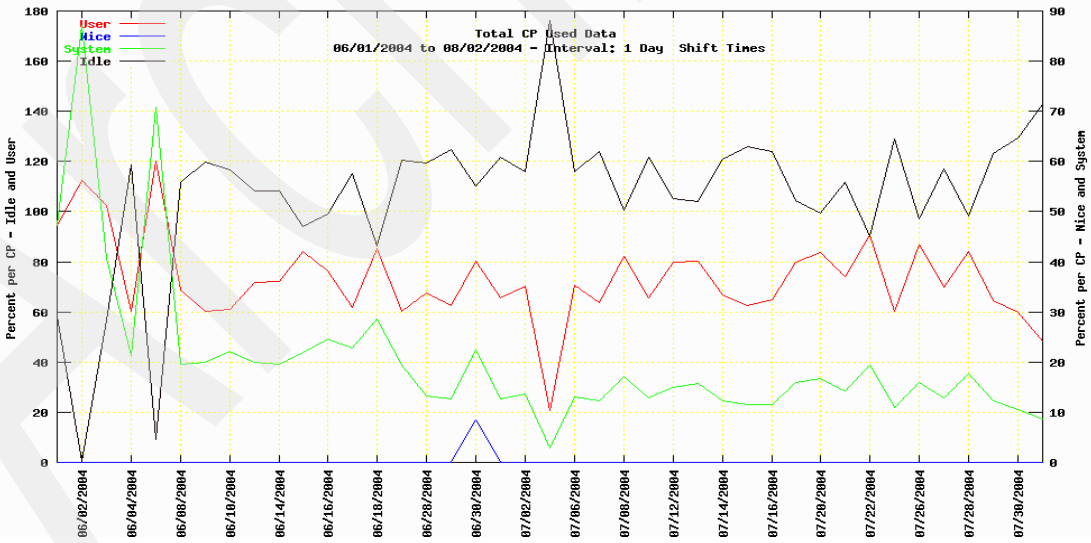



*Figure 12-14 CPU usage in prime shift*

The longer the interval, the less the fluctuation in the data. However, while this is a good capacity chart, we need to remember the higher peaks in the first detailed chart. Planning for the average capacity could led to performance problems, since the extra capacity is not there for the short duration peaks. Domino can have a very dynamic resource requirements because of the nature of the product: a mass mailing, user agents, full text indexing,

replication, or Adminp processing of the name and address book (NAB) are just a few examples of things that can cause a high CPU spike but are normal processing for Domino. It should be noted that not only is CPU affected, but also memory and I/O usage can spike during these times. You will need to understand the difference between your daily and peak workloads to ensure that you have adequate capacity to prevent response time/performance issues.

In addition to CPU, there are many other areas in the SAR data. For instance, you can see how your Linux workload is growing. Figure 12-15 shows the total number of process in the proclist and that this number is growing. However, while the number of processes is increasing, the number that are waiting to run has decreased and stabilized.

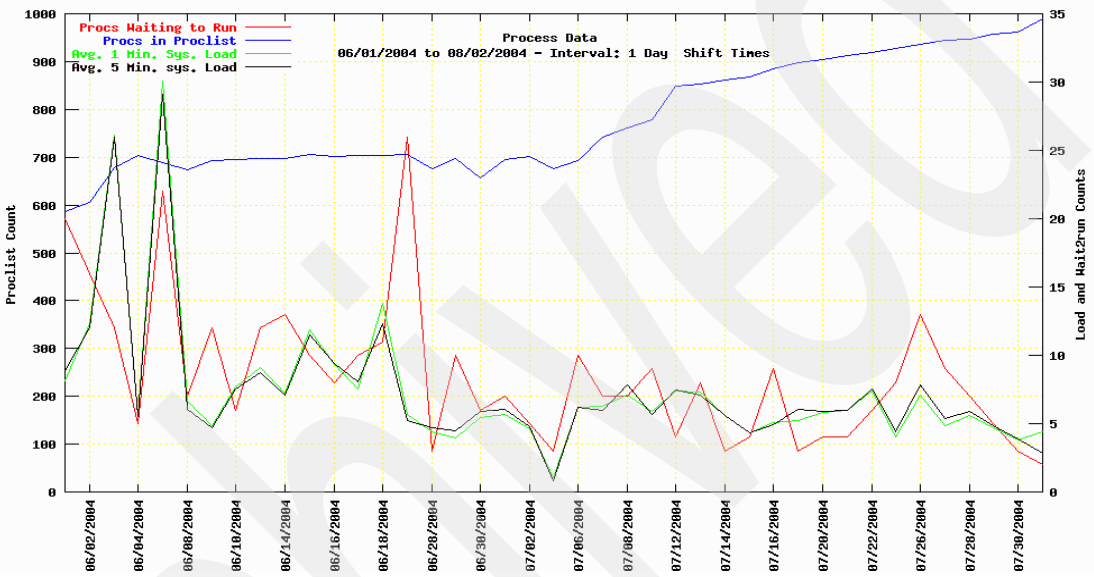



*Figure 12-15 Processes*

You can identify which I/O devices have the most activity with potential bottlenecks. Be careful not to interpret these rates as response times. Even at twice the busy rate, a device on a new SAN could be providing better response time than an older DASD would.

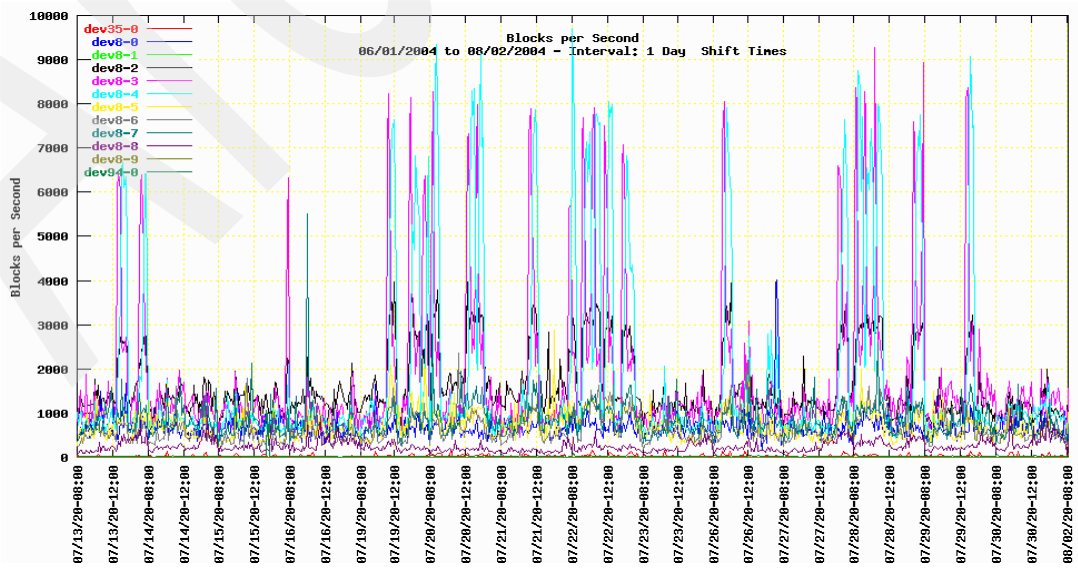



*Figure 12-16 Device activity*

In Linux, the term swapping describes what is called paging in z/OS. While there is a paging statistic in Linux, it deals with the amount of moving of pages to/from the I/O devices, not the movement of pages from memory to the swap file. You will need to monitor the swapping usage in Linux closely. Domino on any platform that is swapping memory to disk will suffer performance degradations. If the swap device is a DASD device, the amount of swapping should be very low to zero. If the swap device is a memory device like XPRAM, the amount of swapping can be larger. However, remember that Domino has a very dynamic and changing workload. If you ration memory to squeeze as much as possible into it and a workload spike occurs (mass mailing or rebuilding indexes on your multi-gigabyte address book, for example), you can push yourself over the memory limit and experience degraded performance.

Figure 12-17 shows a daily example, in a Domino LPAR that is swapping, of the average swapping rates for prime shift. This chart uses page-in and page-out terms, but this means moving pages between storage and auxiliary devices.

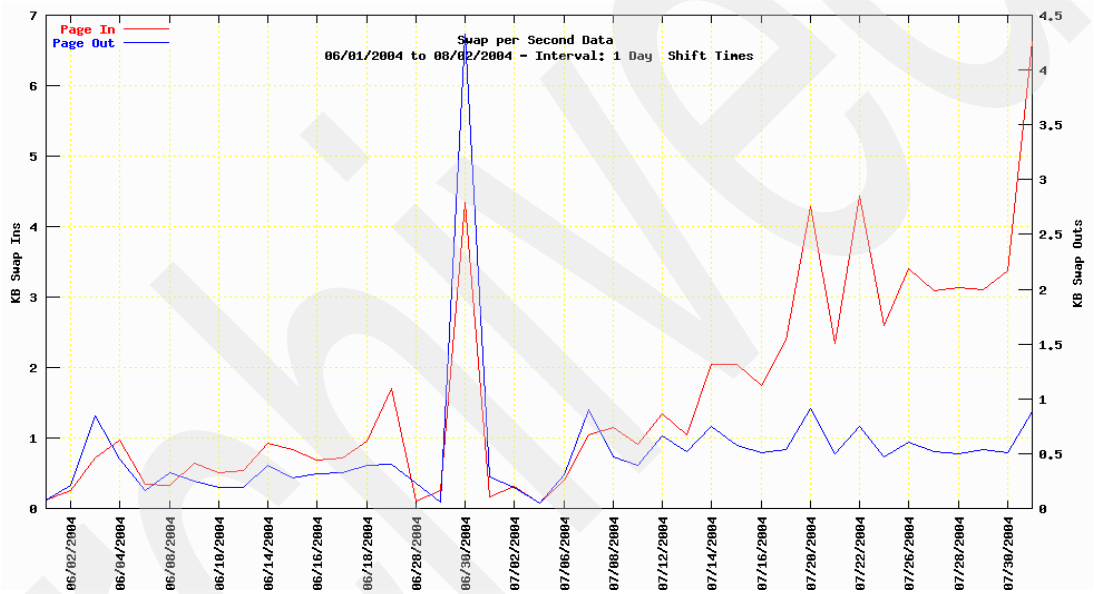



*Figure 12-17 Average swapping rates*

The swap-in rate is a performance indicator while swap-outs are a capacity indicator. Running processes are not delayed by swap-outs; rather, delays occur when a page is no longer in memory and must be swapped back in.

Rates differ depending on types of workloads you are running. For example, a series of serial jobs (Domino examples are Design, Catalog, Updall and other tasks started by the SERVER_AT statements in your notes.ini) might start and run only once in a while. You will see much swap-in activity at the beginning while the tasks are brought into memory, but over time their pages will swap out. As long as they don't run concurrently, you can utilize your memory in series. If you have tasks that run in parallel (such as Domino's Update, Agent Manager, AdminP, Replica, and Cluster tasks) and overlap, you can get a high swap-in rate, as these process are competing for a limited amount of memory. If you have a large number of swap-ins, the solution is not more (or faster) swap space, but more memory. Looking at the same data as in Figure 12-17, but in 10-minute intervals for prime shift (Figure 12-18 on page 328), we see a different picture.

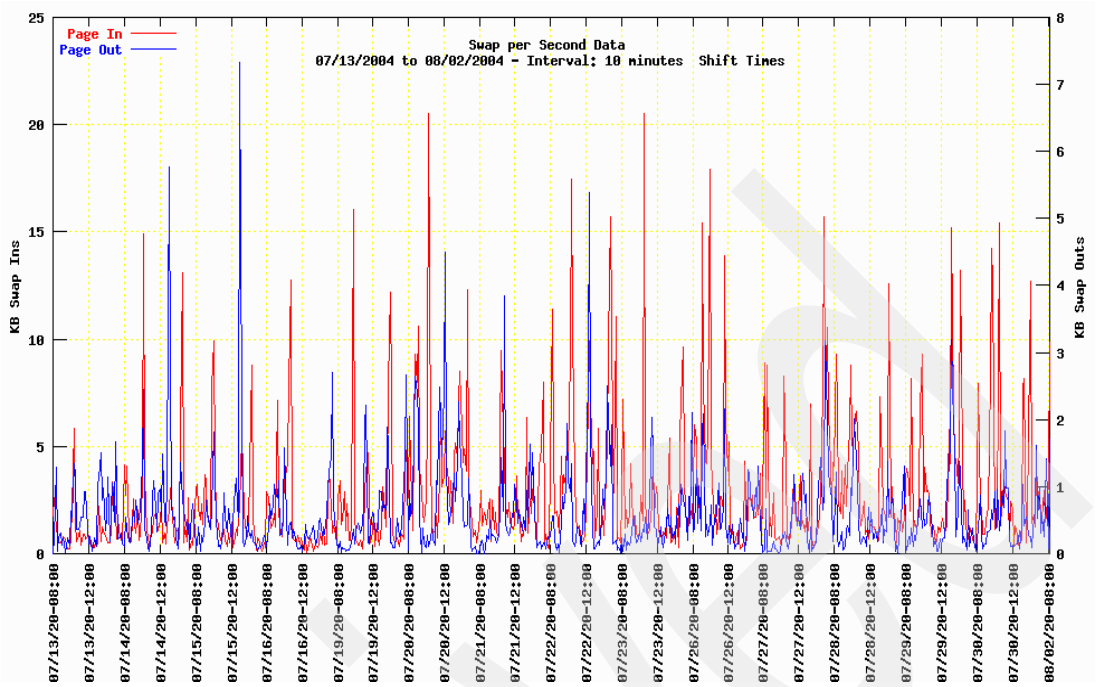



*Figure 12-18 Prime shift swap-ins*

Remember that there are two different Y axis scales. As the average lines start to separate in the daily chart, we can see in the details the very spiky nature as Domino is running its support tasks (Update, Adminp) to your mail workload during prime shift.

There are many other data points in your SAR data that you can become familiar with, in order to understand which one applies to your environment.

## 12.6 Other IBM toolkits

Depending if you are running your Linux machines as guests under z/VM or in native LPARs, there are two options to get additional data. Also, there are other (non-IBM) choices available. You should evaluate your skill set and what each tool provides to decide which tool is best for you.

### 12.6.1 z/VM performance toolkit

z/VM provides a wealth of information about resources the z/VM LPAR is using and what resources each guest machine is using. Figure 12-19 on page 329 shows a sample of the data types that z/VM can capture and feed to the z/VM Performance toolkit.

z/VM Performance Data Food Chain

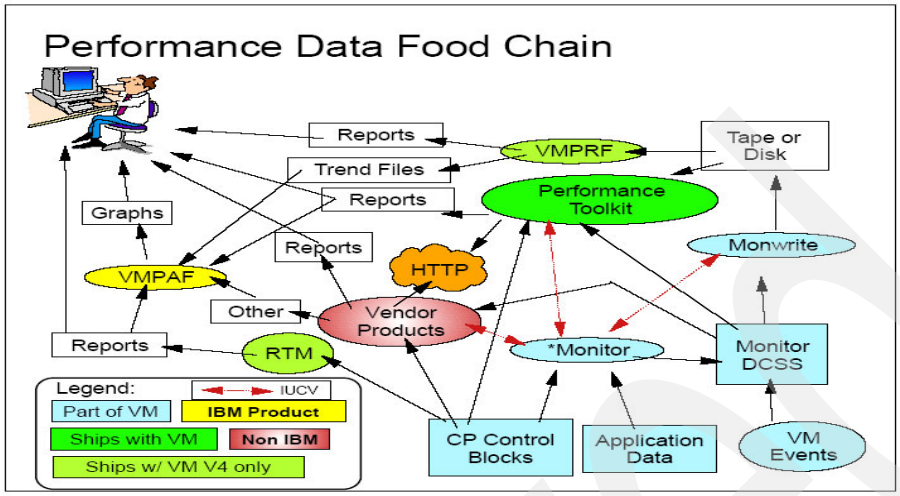



*Figure 12-19 Resources captured by z/VM*

## 12.6.2 RMF/PM

If you are running native Linux LPARs, you can use the RMF/PM toolkit to get information about how your LPARs are running. This requires an z/OS LPAR on the same system, though it doesn't need to be running Domino. The information from the Linux LPARs and z/OS (through the LPAR manager data) is merged down to the graphic RMF/PM tool; see Figure 12-20.

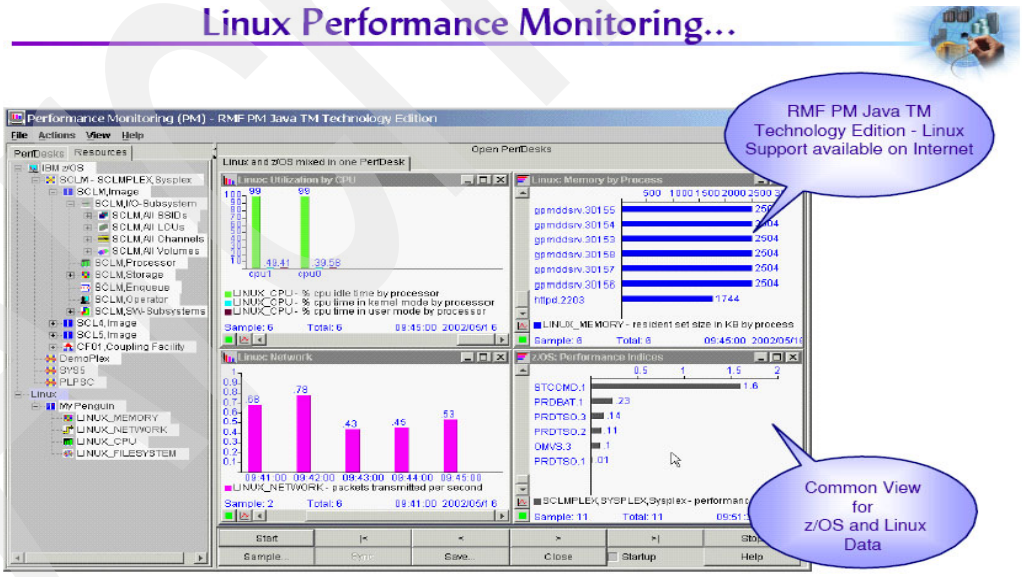



*Figure 12-20 RMF/PM tool UI*

## 12.7 ESALPS overview with guest under z/VM

In developing this redbook, we used ESALPS by Velocity Software to provide performance monitoring tools designed for z/VM and its Linux guests:

ESALPS can be used to monitor VM system performance and analyze bottlenecks. It can be used as a real-time performance monitor, and also for analysis of history and trend files of accumulated performance data.

As a real-time monitor, ESALPS provides displays of VM CPU, LPAR, channel, and I/O performance, as well as resource consumption, response times, and communication rates. When installed and configured for Linux monitoring, ESALPS can report Linux-specific performance data, and also Linux CPU utilization, Linux memory utilization, Linux network activity, and Linux filesystem usage.

ESALPS, the Linux Performance Suite, is a suite of products provided by Velocity Software. The products that make up the suite include:

**ESAMAP** The VM Monitor Analysis Program, providing performance reports on all aspects of VM/ESA and z/VM performance.

**ESAMON** The VM Real Time Monitor, providing real-time analysis of performance.

**ESATCP** The network and Linux data collection program, providing data collection for NT, Unix, and Linux servers, as well as network data.

**ESAWEB** A very fast, VM-based Web server.

In addition to these four products, ESALPS provides a Web-based interface to view performance data through a Web browser and many control facilities.

### 12.7.1 Monitoring requirements

Many requirements for data collection are met by ESALPS. Data is provided for:

- **Capacity planning**
  Long-term data in the form of a performance database (PDB) is needed as input to long-term capacity planning and trend analysis. Full historical data functions are provided with collection and many forms of data extraction tools.
- **Performance analysis**
  Trend data enables an analyst to detect performance changes in any of thousands of potential problem areas. The performance database allows analysts to determine which changes occurred in the system. Reporting on specific periods of time can be performed, enabling an in-depth performance analysis of performance problems.
- **Real-time performance**
  Beyond the traditional “entry level” real-time performance reporting of the top users and system utilization, real-time performance analysis is provided for all subsystems, user activity, and Linux (and many other platforms) servers. Network data is also provided real time.
- **Linux data**
  With the advent of virtual Linux server farms on z/VM, performance data is required.

### 12.7.2 Standard interfaces

ESALPS uses standard interfaces for all data collection. The advantage to using the standard interfaces provided is that when there are a multitude of releases and distributions available, the standard interfaces provide consistent data sources. Supported interfaces include:

- z/VM. A “monitor interface” has been available since 1988. Since then, this interface has provided a consistent view of performance of VM systems.
- Network performance. This is collected using simple network management protocol (SNMP), the standard for network management.
- NETSNMP. An open source software package, it provides host data for Linux and other platforms.
- VM applications. This data interface is used by applications to insert data into the monitor stream consistent with the standard monitor interface. ESATCP uses this interface to ensure consistent data collection that allows full integration of Linux and VM data.

### 12.7.3 Performance database

ESALPS provides both real-time data and historical data for in-depth analysis. The performance data is collected daily with a one-minute granularity based on the monitor interval. A longer term archive is collected, usually with a granularity of 15 minutes. This performance database (PDB) includes VM data, Linux data, and network data.

### 12.7.4 Real-time monitoring with ESAMON

ESAMON uses:

- Historical reporting
- Linux reporting
- Network reporting

Velocity Software has been in business since 1988, supporting the VM environment. With a focus on VM, and now z/VM, Velocity Software added TCP/IP network analysis and then Linux, Microsoft Windows NT®, Sun, and other platforms to the product data collection facilities.

### 12.7.5 Processor

The process for analyzing performance data for the processor should start at a high level. In Figure 12-21 on page 332 you see a simple “SMART” screen (provided by ESAMON) showing the top users and defined servers. The three Linux servers running Domino have similar profiles in terms of CPU and I/O requirements.

Note that this system has sufficient storage, so none of the servers are paging. In this case, the three Linux servers are using almost 70% out of one of the two processors on this system. The percent utilization for a two-processor system is shown up to 200%, to ensure capacity requirements are always comparable across systems

```
TUNER: SMART   ITSO z/Domino Redbook          ESAMON V3.3     08/25 15:09-
 Seconds until next interval: 40                                     2066
   <----------Top Users----------->  <-----------Servers----------->
    Userid:     CPU%  IO/Sec Pg/Sec  Userid:     CPU%  IO/Sec Pg/Sec
 1) LINUXC      29.9   58.71      0  TCPIP        0.2       0      0
 2) LINUXA      20.8   40.04      0  PVM          0.0       0      0
 3) LINUXB      17.8   52.35      0  PVM1         0.0       0      0
 4) ESAWRITE    1.03    1.62      0  RSCS         0.0       0      0
 5) ESATCP      0.88       0      0
 6) SMART       0.23       0      0
 7) SNMPD       0.06       0      0
 8) VTAM        0.00       0      0
 9) NPM         0.00       0      0
10) ESASERVE    0.00       0      0

<---------CPU Statistics-----> <---In Queue User statistics----> <-Page->
%cpu %usr %prb %sys %ovr %idl   InQ   Q0   Q1   Q2   Q3 Eli Ldng <-rate->
72.0 71.1 67.3 0.91 3.76 127.   8.36 1.06 1.13 0.16 6    0    0    0.0
```

*Figure 12-21 ESAMON SMART screen*

Figure 12-22 on page 333 shows a very high level perspective of each of the subsystems. From this, total processor utilization can be read. Given a perspective on total utilization, and knowing what servers are utilizing the processor, the next step is to look at the processes utilizing the resource.

From this display, some general assumptions can be made. Overall, there is CPU left over—the processor is not constrained, since it never approaches 200%. From the paging to expanded storage and paging to DASD, it is obvious that this system has no issues with paging and no further effort in analyzing the paging subsystem is required.

In a Linux server environment, there is not much value in measuring the transaction values provided by z/VM. These are not a measure of work performed inside Linux, but a measure of the whole virtual machine. The storage measure is important in understanding the overall requirements. In this case, there is about 30 MB of overhead to support the users, identified as fixed user storage, and there is about 400 MB of virtual machine storage resident. As this system has 2 GB of real memory, there are no storage constraints.

```
Report: ESASSUM      Subsystem Activity                                Veloc
Monitor initialized:                      on 2066 serial 71CE3     First
----------------------------------------------------------------------
          <---Users----> Transactions <Processor> Storage (MB) <-Paging--
          <-avg number->    Per  Avg. Utilization Fixed Active <pages/sec>
Time       On Actv In Q Minute  Resp Total Virt.  User Resid. XStore DASD
-------- ---- ---- ---- ------ ----- ----- ----- ----- ------ ------ ----
13:51:00   40   21  4.0  164.0 0.187   116   102  30.0  401.1      0    0
13:52:00   40   22  5.0  163.0 0.199   149   130  29.8  401.0      0    0
13:53:00   41   21  7.0  161.0 0.190   145   125  31.5  401.0      0    0
13:54:00   41   21  3.0  161.0 0.203   168   144  29.0  401.0      0    0
13:55:00   41   19  3.0  163.0 0.189   179   153  31.8  401.1      0    0
13:56:00   41   19  5.0  159.0 0.207   187   158  30.2  401.2      0    0
13:57:00   41   22  4.0  165.0 0.193   184   153  28.9  401.1      0    0
13:58:00   41   19  5.0  169.0 0.194   175   142  30.3  401.2      0    0
13:59:00   42   21 10.0  160.0 0.181   178   146  31.6  402.8      0    0
14:00:00   42   22  3.0  172.0 0.197   173   143  32.4  402.4      0    0
**************************************************Summary***************
Average:   42   20  3.1  163.2 8.194    89    77  29.6  408.1      0    0
* * * End of File * * *
```

*Figure 12-22 ESASUM report*

The next step in analyzing performance of a Linux server, now that the basic subsystems have been evaluated, is to measure inside the Linux server. In this example, there is one process that is likely looping. Processes using such large amounts of processor resource should be identified and a determination should be made if it is appropriate or not.

Figure 12-22 also shows the storage requirement of each process. In Linux, there is no distinction between threads and processes and though storage is shared between threads, the reporting of the storage will contain large overlaps.

In Figure 12-23 on page 334, all the processes identified as `server` are sharing storage.

```
Screen: ESAHST1  ITSO z/Domino Redbook          ESAMON V3.3  08/22 14:18-14:19
1 of 1  LINUX HOST Software Analysis Report     NODE * LIMIT 500     2066 71CE3

                  <--Software Program----------> <CPU Seconds> CPU  Storage(K)
Time     Node     Name      ID   Type    Status  Total Intrval Pct   Current
-------- -------- -------- ----- ------- ------- ----- ------- ----- ---------
14:19:00 LINUXC   amgr     27059 Applic  Running   334   58.52 97.67      3896
                  snmpd      298 Applic  Running  1485    1.34  2.25      1768
                  adminp   27219 Applic  ResWait     1    0.34  0.57      6104
                  update   26666 Applic  ResWait     2    0.16  0.26      2296
                  server   26612 Applic  ResWait    34    0.12  0.20     24568
                  java      3818 Applic  ResWait     2    0.12  0.20      6900
                  java     26280 Applic  ResWait    55    0.11  0.18      6900
                  java     26284 Applic  ResWait    55    0.08  0.13      6900
                  java      3819 Applic  ResWait    42    0.08  0.13      6900
                  adminp   26551 Applic  ResWait    21    0.07  0.11      8404
                  update   27285 Applic  ResWait     1    0.06  0.10      2148
                  server   26627 Applic  ResWait    21    0.06  0.10     24568
                  ksoftirq     7 Applic  ResWait    28    0.05  0.08         0
                  server   26527 Applic  ResWait     2    0.04  0.07     24568
                  compact   3265 Applic  ResWait     0    0.04  0.07     42080
                  server   26286 Applic  ResWait     6    0.03  0.05     24568
                  sched    26733 Applic  ResWait     2    0.02  0.03      4256
                  server   26573 Applic  ResWait     1    0.02  0.03     24568
                  server   26571 Applic  ResWait     1    0.02  0.03     24568
                  java     26283 Applic  ResWait     7    0.02  0.03      6900
                  java     26028 Applic  Running     9    0.02  0.03      6900
                  java     26026 Applic  ResWait     8    0.02  0.03      6900
                  amgr     26744 Applic  ResWait     1    0.01  0.02       748
                  server   26621 Applic  ResWait     0    0.01  0.02     24568
                  server   26584 Applic  ResWait     1    0.01  0.02     24568
                  server   26575 Applic  ResWait     1    0.01  0.02     24568
                  server   26558 Applic  ResWait     1    0.01  0.02     24568
                  server   26554 Applic  ResWait     1    0.01  0.02     24568
```

*Figure 12-23 ESAHST1 screen*

## 12.7.6 Linux disk performance

During our project, we experimented with various types of disk and channel configurations, as well as with different formatting methods. In the following sections, we discuss our findings.

### LVM

We can use Linux commands to display our system status. The example in Figure 12-24 on page 335 shows that the LVM is not striped and that the disk initially fills up from the first Physical Volume in the Volume group to the rear.

```
lvdisplay -v /dev/domino/mail1
--- Logical volume ---
LV Name                /dev/domino/mail1
VG Name                domino
LV Write Access        read/write
LV Status              available
LV #                   1
# open                 1
LV Size                16 GB
Current LE             4095
Allocated LE           4095
Allocation             next free
Read ahead sectors     1024
Block device           58:0

   --- Distribution of logical volume on 7 physical volumes  ---
   PV Name                  PE on PV     reads      writes
   /dev/dasdf1              585          1095503    25890
   /dev/dasdg1              585          1054097    19464
   /dev/dasdh1              585          1076301    16486
   /dev/dasdi1              585          908152     27618
   /dev/dasdj1              585          0          0
   /dev/dasdk1              585          0          0
   /dev/dasdl1              585          0          0
```

*Figure 12-24 Using lvdisplay to examine an LVM*

The example in Figure 12-25 shows the same disk, which has just been striped

```
lvdisplay -v /dev/domino/mail1
--- Logical volume ---
LV Name                /dev/domino/mail1
VG Name                domino
LV Write Access        read/write
LV Status              available
LV #                   5
# open                 0
LV Size                13.73 GB
Current LE             3514
Allocated LE           3514
Stripes                7
Stripe size (KByte)    16
Allocation             next free
Read ahead sectors     1024
Block device           58:4

   --- Distribution of logical volume on 7 physical volumes  ---
   PV Name                  PE on PV     reads      writes
   /dev/dasdi1              502          3          8085
   /dev/dasdj1              502          5          8085
   /dev/dasdk1              502          1          8082
   /dev/dasdl1              502          1          8088
   /dev/dasdm1              502          1          8083
   /dev/dasdn1              502          1          8080
   /dev/dasdo1              502          1          8082
```

*Figure 12-25 lvdisplay on a striped LVM*

From a VM perspective, the difference between striping and not striping can also be seen in Figure 12-27 on page 337. In addition, this report shows the difference between ESS and RVA disks.

> **Note:** The first column and the last two columns in Figure 12-27 have been added for clarification in this book and are not a part of the ESADSD2 report from Velocity Software.
>
> The first column represents groups of disks that have the same configuration that we compare against other groups. The last two columns represent what the group is a part of.

In Figure 12-27 on page 337, the first group of disks are RVA disks that are a part of non-striped partition on LinuxC, with a mount point of mail1. The partition is actually made up of seven volumes; however, only the first three were used. This means that only three out of the seven volumes were used. The others were inactive, as there was no data placed on them.

This reduces the number of volumes to write to and causes delays due to disk contention. The RVA disks are using four CHPIDs and are also using ESCON. The ESACHAN report in Figure 12-26 shows that the first four CHPIDs are being used at about 45% utilization, where the ESS is using two CHPIDs with FICON and the utilization is about 18%

The second and third group of disks are RVA disks that have been striped. All disks are being used about the same amount. As in the first case, the response times are higher-up the overall throughput is spread among all disks.

The fourth group of disks are on the ESS, but are non-striped. This shows that the response, pending, and connection times have been changed dramatically with using ESS and FICON, as compared to the first group of disks.

The fifth group of disk is interesting in that there are two RVA disks as a part of the stripe of seven in the disk partition. The overall response, pending, and connection times have been reduced, as well as the remainder of the stripe having very good times.

The sixth group of disks is a complete set of ESS disks which are all a part of a one-disk partition with seven disks. The best times seen for the performance tests were from this DASD.

In the ESACHAN (channel performance analysis) in Figure 12-26, the ESCON channels in use by the RVA are at 44% to 45% utilized. With the FICON channels, which are significantly faster than ESCON channels, there is new measurement technology. This report shows the overall calculated utilization and the activity across the channel in each of several categories.

```
Report: ESACHAN      Channel Performance Analysis                  Domino Redbook                ESAMAP 3.4.0 08/25/03  Page 205
 Monitor initialized:                      on 2066 serial 71CE3     First record analyzed: 08/24/03 18:00:00
 ---------------------------------------------------------------------------------------------------------------------------------
         <Pct Channel>            <---------------Data Units ------------>  <----Work Unit---->  <-Bus Cycles>  Bytes
 Time/    Utilization      Chanl  <---Reads/Second-->  <--Writes/Second-->  <-Rates / Second-->  <-per Second>  /Data
 CHPID    LPAR Total  Shrd Type   LPAR TOTAL Pct  Max  LPAR TOTAL pct  MAX  LPAR TOTAL Pct  MAX  Used pct  Max   Unit
 -------- ---- -----  ---- -----  ---- ----- --- ----  ---- ----- --- ----  ---- ----- --- ----  ---- --- ----  -----
 08/24/03
 20:00:00
 08          .  44.3   Yes ESCON
 09          .  45.1   Yes ESCON
 10          .  45.8   Yes ESCON
 11          .  44.9   Yes ESCON
 14          .  19.0   Yes FICON  2315  2315 2.0 117K   554   554 0.5 117K  5057  5057 3.9 130K  3679 5.7  65K   1024
 15          .  17.7   Yes FICON  2314  2314 2.0 117K   555   555 0.5 117K  5197  5197 4.0 130K  3683 5.7  65K   1024
 -------- ---- -----
 System:      . 228.1
```

*Figure 12-26 ESACHAN report*

```
 Report: ESADSD2      DASD Performance Analysis                    Domino Redbook                ESAMAP 3.4.0
 08/25/03  Page 191
 Monitor initialized:                      on 2066 serial 71CE3     First record analyzed: 08/24/03 18:00:00
 ---------------------------------------------------------------------------------------------
                                                        <--------DASD Response times
 Dev        Device <--SSCH--> <%DevBusy> <SSCH/sec->        <--Service times-->
 No. Serial Type   Total  ERP  Avg  Peak   avg  peak  Resp  Serv Pend Disc Conn DASD
---- ------ ------ ------ ---  ---- ---- ----- ----- -----  ---- ---- ---- ---- ----
  08/24/03
  19:00:00
***Top DASD by Device busy***
1    151E LX151E 3390-3 110855   0  42.6 42.6  30.8  45.5  13.9  13.8  0.8  5.9  7.2  0.0       LINUXC MAIL1
1    151F LX151F 3390-3 101704   2  41.3 41.3  28.3  40.7  14.6  14.6  0.8  6.4  7.4  0.0       LINUXC MAIL1
1    155E LX155E 3390-3  54391   1  24.7 24.7  15.1  29.5  16.4  16.4  0.9  6.1  9.3  0.0       LINUXC MAIL1
     15D9 LX15D9 3390-3  75870   0  20.4 20.4  21.1  38.7   9.7   9.7  1.1  2.7  5.9    0       LINUXC TRANSLOGD
     1599 LX1599 3390-3  73806   0  19.8 19.8  20.5  30.0   9.6   9.6  1.0  2.5  6.1    0       LINUXC TRANSLOGC
     15DC LX15DC 3390-3  65372   0  19.0 19.0  18.2  28.4  10.5  10.5  1.3  2.6  6.6  0.0       LINUXB TRANSLOGB
2    159B LX159B 3390-3  27635   1  13.5 13.5   7.7  11.6  17.6  17.6  2.7  7.4  7.4    0       LINUXB MAIL2
2    15DA LX15DA 3390-3  27694   0  13.4 13.4   7.7  11.1  17.5  17.5  2.8  7.2  7.4    0       LINUXB MAIL2
2    159A LX159A 3390-3  28441   1  13.4 13.4   7.9  11.5  17.0  17.0  2.7  7.0  7.3    0       LINUXB MAIL2
2    159C LX159C 3390-3  27573   1  13.4 13.4   7.7  11.4  17.5  17.5  2.7  7.3  7.5    0       LINUXB MAIL2
2    159D LX159D 3390-3  27252   1  13.3 13.3   7.6  11.2  17.6  17.6  2.7  7.4  7.5    0       LINUXB MAIL2
2    15DB LX15DB 3390-3  27295   0  13.2 13.2   7.6  10.4  17.5  17.5  2.8  7.2  7.5    0       LINUXB MAIL2
2    155D LX155D 3390-3  27131   1  13.1 13.1   7.5  10.7  17.4  17.4  2.7  7.3  7.5    0       LINUXB MAIL2
3    151A LX151A 3390-3  24043   0  11.0 11.0   6.7  10.1  16.5  16.5  2.2  6.8  7.5    0       LINUXB MAIL1
3    155C LX155C 3390-3  23777   0  10.8 10.8   6.6  10.0  16.3  16.3  2.1  6.5  7.7  0.0       LINUXB MAIL1
3    151B LX151B 3390-3  22339   0  10.7 10.7   6.2   9.3  17.2  17.2  2.2  7.1  7.9    0       LINUXB MAIL1
3    155B LX155B 3390-3  22710   0  10.6 10.6   6.3  10.8  16.7  16.7  2.1  6.7  7.9    0       LINUXB MAIL1
3    155A LX155A 3390-3  22299   0  10.4 10.4   6.2   9.8  16.8  16.8  2.1  6.9  7.8    0       LINUXB MAIL1
3    151D LX151D 3390-3  21851   0  10.4 10.4   6.1  10.1  17.1  17.1  2.1  7.1  7.8    0       LINUXB MAIL1
3    151C LX151C 3390-3  22231   0  10.3 10.3   6.2   8.5  16.7  16.7  2.1  6.8  7.8    0       LINUXB MAIL1
4    8026 LX8026 3390-3 121519   0   8.2  8.2  33.8  54.6   2.4   2.4  0.3  1.1  1.0    0       LINUXC MAIL2
4    8027 LX8027 3390-3 119147   0   7.7  7.7  33.1  48.9   2.3   2.3  0.3  1.1  1.0    0       LINUXC MAIL2
     15DF LX15DF 3390-3  20335   0   7.3  7.3   5.7  11.0  12.8  12.8  1.2  4.5  7.1    0       LINUXC DOMSERVC
5    3E44 LX3E44 3390-3  25771   0   4.7  4.7   7.2  10.9   6.6   6.6  0.3  0.2  6.1    0       LINUXA MAIL1
5    3F44 LX3F44 3390-3  24908   0   4.7  4.7   6.9   9.7   6.8   6.8  0.3  0.2  6.3    0       LINUXA MAIL1
     15DD LX15DD 3390-3   7133   0   3.4  3.4   2.0   8.5  17.1  17.1  3.2  4.3  9.7    0       LINUXB DOMSERVB
4    8125 LX8125 3390-3  55937   0   3.4  3.4  15.5  35.0   2.2   2.2  0.3  0.6  1.3    0       LINUXC MAIL2
     8222 LX8222 3390-3  69127   0   2.4  2.4  19.2  29.3   1.3   1.3  0.3  0.1  0.9    0       LINUXA TRANSLOGA
6    8025 LX8025 3390-3  24979   0   2.1  2.1   6.9  10.6   3.1   3.1  0.3  1.6  1.1    0       LINUXA MAIL2
5    8021 LX8021 3390-3  24530   0   2.1  2.1   6.8   9.7   3.0   3.0  0.3  1.6  1.1    0       LINUXA MAIL1
5    8023 LX8023 3390-3  23956   0   2.1  2.1   6.7  10.1   3.1   3.1  0.3  1.6  1.1    0       LINUXA MAIL1
5    8022 LX8022 3390-3  23994   0   2.1  2.1   6.7   9.4   3.1   3.1  0.3  1.6  1.1    0       LINUXA MAIL1
5    8020 LX8020 3390-3  23599   0   2.0  2.0   6.6   9.1   3.1   3.1  0.3  1.7  1.1    0       LINUXA MAIL1
5    8024 LX8024 3390-3  23888   0   2.0  2.0   6.6  10.7   3.1   3.1  0.3  1.6  1.1    0       LINUXA MAIL1
6    8220 LX8220 3390-3  23970   0   1.9  1.9   6.7  10.7   2.9   2.9  0.3  1.4  1.2    0       LINUXA MAIL2
6    8124 LX8124 3390-3  25081   0   1.1  1.1   7.0   9.9   1.6   1.6  0.3  0.1  1.2    0       LINUXA MAIL2
6    8121 LX8121 3390-3  25174   0   1.1  1.1   7.0  10.2   1.6   1.6  0.3  0.1  1.2    0       LINUXA MAIL2
6    8123 LX8123 3390-3  24725   0   1.1  1.1   6.9   9.8   1.7   1.7  0.3  0.1  1.2    0       LINUXA MAIL2
6    8122 LX8122 3390-3  24327   0   1.1  1.1   6.8  10.2   1.7   1.7  0.3  0.1  1.2    0       LINUXA MAIL2
6    8120 LX8120 3390-3  24233   0   1.1  1.1   6.7   9.8   1.7   1.7  0.3  0.1  1.2    0       LINUXA MAIL2
     1519 LX1519 3390-3   3814   0   0.8  0.8   1.1   2.3   7.5   7.5  1.6  2.8  3.2    0       LINUXC ROOT
     8227 LX8227 3390-3  17352   0   0.7  0.7   4.8  10.3   1.6   1.6  0.3  0.2  1.1    0       LINUXC DOMSERVC
     8221 LX8221 3390-3  11201   0   0.6  0.6   3.1   9.7   1.9   1.9  0.3  0.2  1.3    0       LINUXA DOMSERVA
***End Top DASD by Device busy***
```

*Figure 12-27 ESADASD2 report*

This data shows several different aspects of performance that you should evaluate when working to improve your DASD performance. DASD response time has four components (for this discussion): queue time, disconnect time, pend time, and connect time. Because of the different technologies used for this DASD, this report demonstrates most of the possible

performance benefits that can be obtained. Note that the RVA response time is in the 16 ms range, and the ESS is in the 2 ms range.

Starting with *connect time*, the ESS uses FICON channels that have the capability of over 100 MB per second—the RVA was on ESCON channels at 17 MB per second. On the ESCON channels, connect time accounted for about 7ms per I/O; this decreased to just over 1ms on the ESS with FICON channels. This was one measure of the improvement and the source of one of the improvements.

*Disconnect time* has more to do with the caching function in the storage controller, given equivalent DASD workloads. The ESS shows consistently improved disconnect times, though it was not quite as static as the connect time.

*Pend time* is the time to start the I/O. Given that the ESCON had four channels to the RVA, this could have been expected to be quite good, as compared to just the two channels to the ESS. But the pend time dropped from 1 to 2 ms down to .3 ms. This could be because ESCON is slower than FICON, so each I/O kept the channel busy for a longer period of time (as seen in connect time), and therefore chance of having a busy channel (and needing to wait) increased.

With the additional software options using LVM, you should now be able to understand what technology would apply to fit your DASD I/O needs.

### 12.7.7 Swapping (Linux)

Linux swapping is not true swapping; the term refers to a time in the past when Linux used to swap out an entire process when it was not in use so that another running process could have its memory. This was a very expensive way of managing memory, because there was a big hit on the amount of context switching that occurred. The current swap algorithm uses a paging mechanism. This means that only those pages in memory that are no longer in use will be swapped out to the swap device, rather than the entire process.

The kswapd daemon takes any page that has been marked as *dirty* and swaps it out to the swap device. The kswapd daemon wakes up during two time frames: every 5 minute, to check the dirty list to free up memory, and on demand, each time a process uses more than the last 20% of memory. When a process uses memory that is in the last 20%, kswapd will run to take any pages that have been marked as being dirty and write those pages out to the swap device to get memory use down below 80%.

On VM, we have the advantage of a very fast device that can be used as a swap device: a VM virtual disk, which is essentially a RAM disk that can be defined for the Linux guest. This reserves the amount of swap space that the Linux guest might require, but does not take all of the disk's amount of central storage until it is needed.

This means that the virtual disks are ready to go but are not consuming central storage until Linux starts swapping. Should there be different amounts of swapping, it is best to have multiple virtual disks defined as swap devices, as only the virtual disks that are used will take up central storage. Any amount of swap space that is not used will not tie up central storage (except for the small amount that the virtual disk itself is not using).

For additional discussion on this topic and recommendations on using Linux swapping and virtual disk swapping, see 11.12, "Linux swapping" on page 312.

As we lowered the available virtual and overall storage, processor requirements to support swapping to the virtual disk increased. This has a cost: in our measurement of the 256 MB virtual machine experiment with a 1500-user load, swapping to virtual disk was about 1000

per second; see Figure 12-28. The number of virtual disk I/Os is the measurement for swapping to virtual disk. For the four 15-minute intervals reported, there was an average of 850 K virtual disk I/Os, or almost 1000 per second over the 900-second reporting interval.

```
Report: ESAUSR3      User Resource Utilization - Part 2 Domino Redbook ESAMAP 3.4.0 08/25/03  Page 217
 Monitor initialized: on 2066 serial 71CE3 First record analyzed: 08/21/03 12:00:00
-----------------------------------------------------------------------------------------------
                 DASD MDisk Virt Cache I/O    <---Virtual Device---->
 UserID    DASD Block Cache Disk   Hit Prty   <----I/O Requests----->
 /Class     I/O   I/O  Hits  I/O   Pct Queued  Cons   U/R  CTCA Other
 -------- ----- ----- ----- ---- ------ ----- ----- ----- ----- -----
 08/21/03
 12:15:00  613K     0  248K 838K  74.8      0  1510     0   321     0
***Top User Analysis***
 LINUXA    610K     0  246K 838K  74.8      0     1     0     0     0
-------------------------------------------------------------------------
 12:30:00  615K     0  250K 822K  74.6      0  1487     0   324     0
***Top User Analysis***
 LINUXA    613K     0  248K 822K  74.6      0     0     0     0     0
-------------------------------------------------------------------------
 12:45:00  631K     0  260K 884K  75.5      0  1634     0   321     1
***Top User Analysis***
 LINUXA    628K     0  258K 884K  75.5      0     0     0     0     0
```

*Figure 12-28 ESAUSR3 report showing virtual disk I/O for VM guest for 196 MB run*

The cost of performing this activity is charged to the kswapd daemon. In looking at the Host Application report in Figure 12-29 on page 340, the cost is about 9% of an engine to perform nearly 1000 swap I/Os per second. In addition to the Linux swapping cost, the CPU usage of VM should be added in.

```
Report: ESAHSTA   LINUX HOST Application Report    Domino Redbook  ESAMAP 3.4.0 08/25/03  Page 298
Monitor initialized:    on 2066 serial 71CE3 record analyzed: 08/21/03 12:00:00
--------------------------------------------------------------------------------
 Node/    Process/    <-Application Process Counts------> <-----Processor---->
 Date     Application                                     <---Utilization---->
 Time     name        Total active Running ResWait Loaded Percent seconds Avg
 -------- ----------- ----- ------ ------- ------- ------ ------- ------- ----
 08/21/03
 12:15:00
 LINUXA
          java         15.0   15.0     2.0    13.0      0    10.3    92.6  0.7
          kswapd        1.0    1.0       0     1.0      0     9.1    82.2  9.1
          router       11.0   11.0       0    11.0      0    10.6    95.4  1.0
          server       67.0   67.0     1.0    63.0    3.0    63.2   568.5  0.9
          snmpd         1.0    1.0     1.0       0      0     3.3    29.3  3.3
          update        3.0    3.0     1.0     2.0      0    10.2    91.7  3.4
--------------------------------------------------------------------------------
 12:30:00
 LINUXA
          java         17.0   17.0     2.0    15.0      0     9.5    85.9  0.6
          kswapd        1.0    1.0       0     1.0      0     8.8    79.5  8.8
          router       12.0   12.0     2.0     9.0    1.0    11.0    99.3  0.9
          server       61.0   61.0     4.0    55.0    2.0    62.7   563.9  1.0
          snmpd         1.0    1.0     1.0       0      0     3.2    28.8  3.2
          update        4.0    4.0       0     4.0      0    12.0   107.8  3.0
--------------------------------------------------------------------------------
 12:45:00
 LINUXA
          java         16.0   16.0       0    16.0      0    10.3    92.4  0.6
          kswapd        1.0    1.0       0     1.0      0     9.5    85.6  9.5
          router       10.0   10.0       0    10.0      0    11.1    99.6  1.1
          server       67.0   67.0     9.0    53.0    5.0    64.3   578.6  1.0
          snmpd         1.0    1.0     1.0       0      0     2.4    21.9  2.4
          update        5.0    5.0       0     5.0      0    13.0   116.9  2.6
```

*Figure 12-29 ESAHSTA report for 196 MB run showing kswapd CPU requirement*

The goals of the follow-on experiments were to reduce the machine size first to 196 MB, then to 128 MB. The run with 196 MB provided equivalent response time, with the swap rate averaging about 10% higher, and the kswapd daemon also about 10% higher.

The 128 MB experiment proved the case that the swap rate is linearly proportional to the CPU required by the kswapd daemon. At the one complete interval shown, swapping was 4 million for the interval, or over 4000 per second. The processor utilization was close to 200% out of the two processors, as compared to about 130% out of two processors for the previous run.

```
Report: ESAUSR3  User Resource Utilization - Part 2    Domino Redbook   ESAMAP 3.4.0 08/25/03  Page 222
 Monitor initialized:     on 2066 serial 71CE3     First record analyzed: 08/21/03 15:30:00
 ---------------------------------------------------------------------------------------------------
                 DASD MDisk Virt Cache I/O    <---Virtual Device---->
 UserID    DASD Block Cache Disk   Hit Prty   <----I/O Requests----->
 /Class     I/O   I/O  Hits  I/O   Pct Queued  Cons   U/R  CTCA Other
 -------- ----- ----- ----- ---- ------ ----- ----- ----- ----- -----
 08/21/03
 15:45:00  220K     0  140K 133K  77.5      0  1039     0   322     1
***Top User Analysis***
 LINUXA    215K     0  136K 133K  77.4      0     0     0     0     0
-----------------------------------------------------------------------
 16:00:00  825K     0  581K   4M  94.5      0  1139     0   323     0
***Top User Analysis***
 LINUXA    817K     0  574K   4M  94.5      0     2     0     0     0
-----------------------------------------------------------------------
 16:15:00  354K     0  257K   2M  95.9      0  1037 13525   321     0
***Top User Analysis***
 LINUXA    351K     0  256K   2M  95.9      0     0     0     0     0
```

*Figure 12-30 ESAUSR3 showing Virtual Disk I/O 128 MB run*

```
Report: ESAHST1  LINUX HOST Software Analysis Report   Domino Redbook   ESAMAP 3.4.0 08/25/03  Page 277
 Monitor initialized:      on 2066 serial 71CE3    First record analyzed: 08/21/03 15:30:00
 ---------------------------------------------------------------------------
 Node/    <-----Software Program-------> <CPU Seconds> CPU  StgSize
 Time     Name      ID    Type   Status  Total Intrval Pct  (Bytes)
 -------- --------  ----  ------ ------- ----- ------- ---- -------
 08/21/03
15:45:00
 LINUXA   kswapd       8  Applic ResWait  20.4    18.4  2.0       0
-----------------------------------------------------------------------
 16:00:00
 LINUXA   kswapd       8  Applic ResWait 208.9   181.6 20.1       0
-----------------------------------------------------------------------
 16:15:00
 LINUXA   kswapd       8  Applic ResWait 273.7    63.7 10.7       0
```

*Figure 12-31 ESAHST1 Process Analysis showing kswapd CPU for 128 MB run*

There is a trade-off between reducing storage and the CPU requirement for swapping. Too much swapping increases CPU, but defining too much storage for a virtual machine will increase the overall real storage requirement. This will be a constant area for analysis and planning in order to utilize your current resources most effectively.

### 12.7.8 SRM parameters

To see the effect of changing SRM parameters in VM, see topic 6.4.1 in *Linux on IBM @serverzSeries and S/390: Performance Measurement and Tuning*, SG24--6926, which contains a detailed description of the different SRM parameters and how they affect VM.

## 12.8 VM case studies

In this section, we include two practical examples for using reported data to solve a problem.

## 12.8.1 Linux console goes to HOLDING state

Suppose you have the Linux console is logged on and a terminal is attached to the console. If the terminal is in HOLDING mode and the CP run option is OFF (which is the default), all processing will be stopped until the screen is cleared. As shown in Figure 12-32, there is a gap in the data received, because the processing for this VMID has been suspended until the Hold was cleared. The hold started in the 15:48 minute and lasted till 15:54. A look at any other screen, including Linux reports, will show that nothing was happening in either VM or Linux for this Linux guest.

This is important to know, as this is something that can happen when the Linux console is a 3270 terminal.

To correct this problem, the CP SET RUN ON command can be issued in the Linux VM ID profile. SET RUN ON prevents the virtual machine from going into CP READ when the virtual console is reconnected.

If you want to prevent the screen from going into HOLDING, you can use the TERM settings. However, if you set TERM MORE 0, you may be unable to enter commands on the tn3270 session when a runaway process floods the console. TERM HOLD OFF may be a good command to issue also.

```
Screen: ESAUSR2  ITSO z/Domino Redbook          ESAMON V3.3  08/19 15:42 -16:00
 1 of 3  User Resource Utilization               USER linuxa          2066 71CE3

                  <---CPU time--> <--------Main Storage (pages)--------->
          UserID   <(seconds)> T:V <Resident>  Lock <-----WSS----->
 Time     /Class   Total  Virt Rat Total Activ  -ed Total Actv  Avg Resrvd
 -------- -------- ----- ----- --- ----- ----- ---- ----- ---- ---- ------
 16:00:00 LINUXA   20.64 19.65 1.1 65492 65492 1419 64008  64K  64K     32
 15:59:00 LINUXA   26.41 25.24 1.0 65492 65492 1147 63896  63K  63K     32
 15:58:00 LINUXA   26.61 25.55 1.0 65492 65492 1131 63929  63K  63K     32
 15:57:00 LINUXA   32.09 30.93 1.0 65492 65492 1995 63449  63K  63K     32
 15:56:00 LINUXA   30.01 28.84 1.0 65492 65492 1035 63703  63K  63K     32
 15:55:00 LINUXA   31.30 30.00 1.0 65492 65492 1291 63785  63K  63K     32
 15:54:00 LINUXA   23.82 22.82 1.0 65492 65492 1339 63465  63K  63K     32
 15:48:00 LINUXA    3.78  3.62 1.0 65492 65492 1148 64296  64K  64K     32
 15:47:00 LINUXA   27.72 26.60 1.0 65492 65492 1371 63400  63K  63K     32
 15:46:00 LINUXA   24.31 23.12 1.1 65492 65492 1211 63496  63K  63K     32
 15:45:00 LINUXA   27.67 26.24 1.1 65492 65492 1996 63896  63K  63K     32
 15:44:00 LINUXA   23.82 22.44 1.1 65492 65492 2027 64361  64K  64K     32
 15:43:00 LINUXA   39.88 37.99 1.0 65492 65492 1260 63865  63K  63K     32
                            PF6= Percent Resource Use
PF1=Help      PF2=Zoom     PF3=Quit  PF4=Select  PF5=Plot               PA1=CP
PF7=Backward  PF8=Forward  PF9=Sort  PF10=Parms  PF11=More   PF12=Exit  PA2=Copy
 ====>
```

*Figure 12-32 Showing a gap in data*

## 12.8.2 DASD hot spot

When a DASD has a “hot spot” (that is, a great deal of I/O to the same VM volume), it can be difficult to trace the VM volume to a Linux mount point to determine what can be done to resolve the issue. In this section, we explain how you can use ESAMON to track down a hot spot.

Use the ESAMON ESADSD2 screen to locate a hot spot on the disk. We found one on the 8220 device. Using the zoom PF key, the screen in Figure 12-33 on page 343 was produced.

Device 8222 was where most of the I/O was taking place. By using the split function and going to the ESAUSEK screen, we can see that the 8222 has a vaddr of 211 and is owned by LinuxA.

```
 Screen: ESADSD2A ITSO z/Domino Redbook          ESAMON V3.3  08/20 13:26-13:27
 1 of 3  DASD Performance Analysis - Part 1      DEVICE 0000-FFFF     2066 71CE3


           Dev        Device %Dev <SSCH/sec-> <-----Response times (ms)--->
 Time      No. Serial Type   Busy   avg  peak  Resp  Serv  Pend  Disc  Conn
 -------- *--- ------ ------ ---- *---- ----- ----- ----- ----- ----- -----
 13:27:00 8220 LX8220 3390-3  0.8   2.9  10.4   2.7   2.7   0.3   0.9   1.5
          8221 LX8221 3390-3  0.4   2.2   6.1   1.7   1.7   0.3   0.0   1.4
          8222 LX8222 3390-3  3.5  32.8  56.8   1.1   1.1   0.3   0.1   0.7
          8223 LX8223 3390-3    .     .     .     .     .     .     .     .
          8224 LX8224 3390-3    .     .     .     .     .     .     .     .
          8225 LX8225 3390-3    .     .     .     .     .     .     .     .
          8226 LX8226 3390-3    .     .     .     .     .     .     .     .
          8227 LX8227 3390-3    .     .     .     .     .     .     .     .
          8228 LX8228 3390-3    .     .     .     .     .     .     .     .


PF1=Help                  PF3=Quit  PF4=Select  PF5=Plot    PF6=Seeks  PA1=CP
PF7=Backward  PF8=Forward  PF9=Sort  PF10=Parms  PF11=More   PF12=Exit  PA2=Copy
 ====>
 Screen: ESAUSEK  ITSO z/Domino Redbook          ESAMON V3.3  08/20 13:26-13:27
 1 of 2  User MDISK Seeks Report                 USER * MDISKS        2066 71CE3

                       Dev. Volume <Cylinder >         Total
 Time     Owner    Vadr No.  Serial Start Stop  Userid   Seeks
 -------- -------- ---- ---- ------ ----- ----- -------- -----
 13:27:00 LINUXA   0211 8222 LX8222     7  3154 LINUXA    1443
 13:27:00 LINUXA   0203 3F44 LX3F44     7   585 LINUXA     247
 13:27:00 LINUXA   0206 8022 LX8022     7   585 LINUXA     218
 13:27:00 LINUXA   0208 8024 LX8024     7   585 LINUXA     205
 13:27:00 LINUXA   0207 8023 LX8023     7   585 LINUXA     192
 13:27:00 LINUXA   0204 8020 LX8020     7   585 LINUXA     192
 13:27:00 LINUXA   0202 3E44 LX3E44     7   585 LINUXA     179
 13:27:00 LINUXA   020A 8120 LX8120     7  2131 LINUXA     171
 13:27:00 LINUXA   020C 8122 LX8122     7  2131 LINUXA     162
 13:27:00 LINUXA   0205 8021 LX8021     7   585 LINUXA     156
 13:27:00 LINUXA   0209 8025 LX8025     7  2131 LINUXA     147
                          PF6= Percent Resource Use
PF1=Help      PF2=Zoom     PF3=Quit  PF4=Select  PF5=Plot               PA1=CP
PF7=Backward  PF8=Forward  PF9=Sort  PF10=Parms  PF11=More   PF12=Exit  PA2=Copy
 ====>
```

*Figure 12-33 ESADSD2 and ESAUSEK screens*

Now that the specific machine and disk are known, the actual Linux filesystem can be found. The first step is to find out what the Linux device name is for the VM minidisk. This is done by looking at the DASD device table to identify the device. The way to look at the DASD device table is to use the Linux **`cat /proc/dasd/devices`** command shown in Figure 12-34 on page 344. The output from this command shows that vaddr 211 is dasdr; in Linux terms, this is /dev/dasdr.

```
linuxa:~ # cat /proc/dasd/devices
0200(ECKD) at ( 94:  0) is dasda     : active at blocksize: 4096, 600840 blocks, 2347 MB
0201(ECKD) at ( 94:  4) is dasdb     : active at blocksize: 4096, 600840 blocks, 2347 MB
0202(ECKD) at ( 94:  8) is dasdc     : active at blocksize: 4096, 600840 blocks, 2347 MB
0203(ECKD) at ( 94: 12) is dasdd     : active at blocksize: 4096, 600840 blocks, 2347 MB
0204(ECKD) at ( 94: 16) is dasde     : active at blocksize: 4096, 600840 blocks, 2347 MB
0205(ECKD) at ( 94: 20) is dasdf     : active at blocksize: 4096, 600840 blocks, 2347 MB
0206(ECKD) at ( 94: 24) is dasdg     : active at blocksize: 4096, 600840 blocks, 2347 MB
0207(ECKD) at ( 94: 28) is dasdh     : active at blocksize: 4096, 600840 blocks, 2347 MB
0208(ECKD) at ( 94: 32) is dasdi     : active at blocksize: 4096, 600840 blocks, 2347 MB
0209(ECKD) at ( 94: 36) is dasdj     : active at blocksize: 4096, 600840 blocks, 2347 MB
020a(ECKD) at ( 94: 40) is dasdk     : active at blocksize: 4096, 600840 blocks, 2347 MB
020b(ECKD) at ( 94: 44) is dasdl     : active at blocksize: 4096, 600840 blocks, 2347 MB
020c(ECKD) at ( 94: 48) is dasdm     : active at blocksize: 4096, 600840 blocks, 2347 MB
020d(ECKD) at ( 94: 52) is dasdn     : active at blocksize: 4096, 600840 blocks, 2347 MB
020e(ECKD) at ( 94: 56) is dasdo     : active at blocksize: 4096, 600840 blocks, 2347 MB
020f(ECKD) at ( 94: 60) is dasdp     : active at blocksize: 4096, 600840 blocks, 2347 MB
0210(ECKD) at ( 94: 64) is dasdq     : active at blocksize: 4096, 600840 blocks, 2347 MB
0211(ECKD) at ( 94: 68) is dasdr     : active at blocksize: 4096, 600840 blocks, 2347 MB
```

*Figure 12-34 cat /proc/dasd/devices*

From this point we can look at the filesystem table in /etc/fstab to see if /dev/dasdr is defined in it. If it is, then the search for the mount point is complete. The fstab would indicate where the device is mounted, and a guess as to what it is being used for can be made.

However, in this case the device is *not* in the filesystem table. So the next place to look is in the Logical Volume Manager to find out which logical volume the device is assigned to. This can be somewhat challenging, as the device can be used in many different logical volumes.

To find out where the device is in the logical volume, the lvdisplay command is used, as shown in Figure 12-35 on page 345. From this command, we can tell the following information:

- We know from the ESAMON ESADSD2 screen in Figure 12-33 on page 343 that the device is a 3390-3. This means that it has 585 physical extents (PEs).
- The lvdisplay command shows that /dev/dasdr is used in the /dev/domino/transloga logical volume.
- It also shows that all 585 physical extents are being used by this logical volume, and that there are no other devices used in this volume.

In this case, we were fortunate that the logical volume maps very easily to the physical device; we had only a single logical volume.

```
linuxa:~ # lvdisplay -v /dev/domino/* | less
:
:
--- Logical volume ---
LV Name                /dev/domino/transloga
VG Name                domino
LV Write Access        read/write
LV Status              available
LV #                   4
# open                 1
LV Size                2.29 GB
Current LE             585
Allocated LE           585
Allocation             next free
Read ahead sectors     1024
Block device           58:3

   --- Distribution of logical volume on 1 physical volume  ---
   PV Name                  PE on PV     reads      writes
   /dev/dasdr1              585          187985     1805150
:
:
```

*Figure 12-35 lvdisplay command*

The final step is to return to the filesystem table (/etc/fstab) and see where the logical volume is mounted. As shown in Figure 12-36, the logical volume with /dev/dasdr (LinuxA's 211 minidisk) is mounted as /domserva/notesdata/translog.

From the name of the mount point, a guess can be made as to the purpose of the device and to why the I/O rate is high. In this case, the mount point is exactly as it has been described, and it is the transaction log disk for Domino.

```
linuxa:~ # cat /etc/fstab
/dev/dasda1          /                    ext3       defaults              1 1
/dev/dasdb1          /opt                 reiserfs   defaults              1 2
/dev/domino/domserva /domserva            ext2       defaults              0 2
/dev/domino/transloga /domserva/notesdata/translog  ext2 defaults          0 2
/dev/domino/mail1    /notesmail/nbmail1   ext2       defaults              0 2
/dev/domino/mail2    /notesmail/nbmail2   ext2       defaults              0 2
devpts               /dev/pts             devpts     mode=0620,gid=5       0 0
proc                 /proc                proc       defaults              0 0
/dev/dasdu1          swap                 swap       defaults              0 0
/dev/dasdv1          swap                 swap       defaults              0 0
```

*Figure 12-36 Filesystem table*

## 12.9 What do I monitor?

This leads to the next question: what should I monitor? Unfortunately, there is no easy answer here. Are you running a mail server or an application server? Is it Notes/NRPC based or Web/HTTP based? Do you replicate or cluster? Do you push changes out to the address book during prime shift? These are just a few examples of ways to run Domino, but each would require the examination of different statistics.

Since rates are an indicator of capacity and not of performance, items like CPU busy do not tell us how well the server is running. Is that spike a normal workload spike (mass mailing, agents, or indexing) or is that spike a real performance issue (not costing you 10x CPU per message to deliver in the router)? Also a CPU drop can be just a deadly to your end users response times. This could indicate a bottleneck that is preventing work from reaching the server or executing on it.

I/O rates do not tell you how well a device is responding to the workload. If you have two devices, one with twice the amount of I/O as the other, which one should you work on? If the device with twice the workload is averaging 2 ms response times and the other is averaging 10 ms, then you would work on the device with the worse response times. Fixing the device with the heavier workload and reducing the response time to 1 or 1.5 ms will not give you as good return on your time as working on the 10 ms device and reducing it to 2 ms.

Response times and cost are good indicators of performance. Since Domino workload are dynamic, the raw CPU numbers will change in a dynamic way. Figure 12-37 shows the total CPU used by each Domino partition running in this LPAR. If we look at the red line (D01ML083), we can see that the CPU cost is relatively flat with a jump around September 6th. On this date, a change was made to this server. By looking at this CPU cost, we would say that this change didn't affect the server, or perhaps increased it slightly.

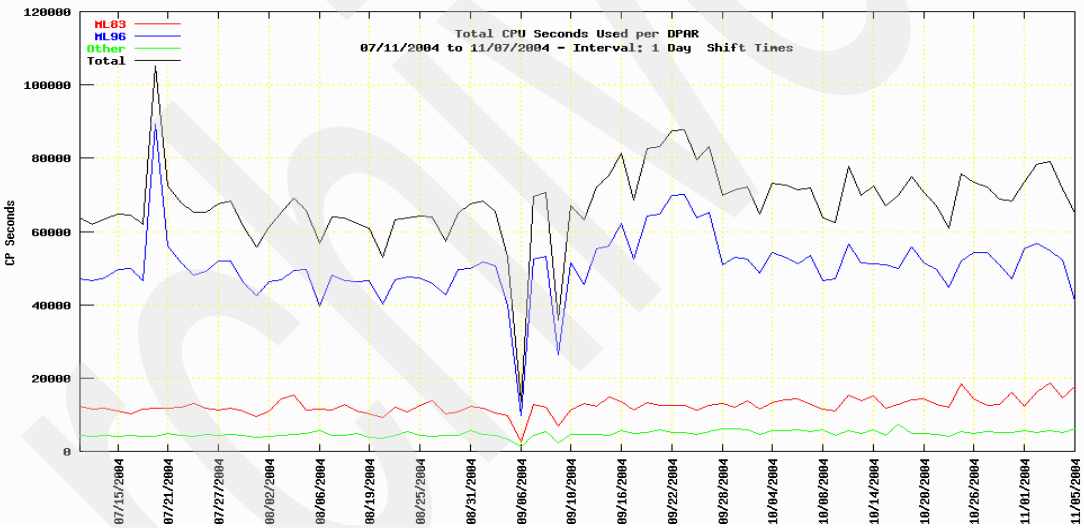



*Figure 12-37 Total CPU used by each Domino server*

If we take the same numbers divide them by the active 15 user counts, we get a very different chart; see Figure 12-38 on page 347.

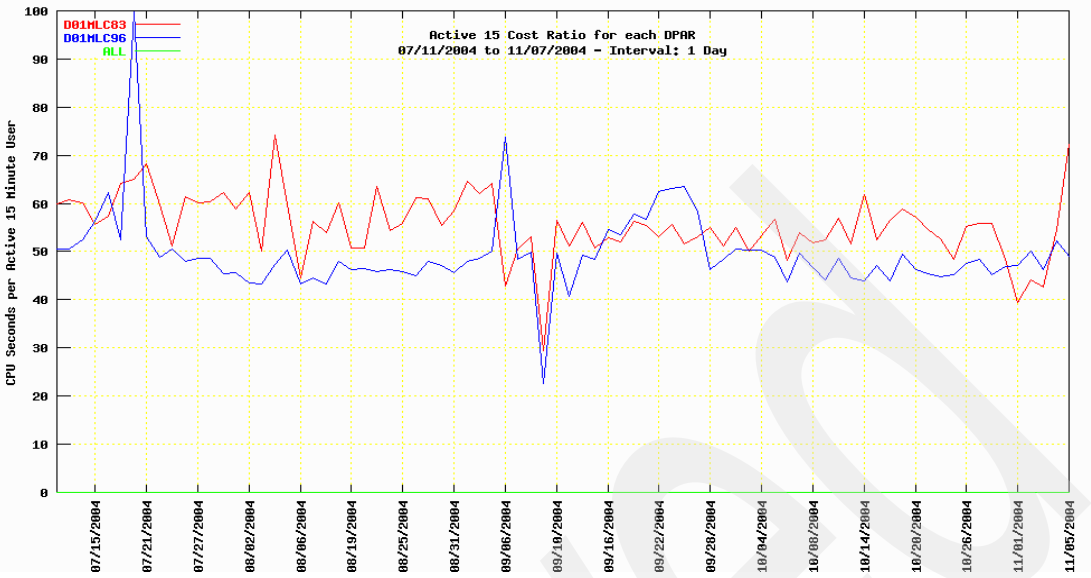



*Figure 12-38 Cost per active user*

Figure 12-38 shows that the change made around September 6th had a good impact on the server. How can this be if the CPU went up? Since this chart is based on cost per unit (per active user in this case), we can determine that the difference between the two servers is more users. Since the cost per user dropped, but the total number users on this server went up, the raw CPU requirements went up. This server is part of a cluster. When the other side was down, more users came over and stayed on this side the cluster. Rates cannot tell you about performance in a Domino environment. The only Domino environment where rates can be used as a performance indicator is a benchmark where you can control the rate and flow of work into the server.

Each Domino setup is unique, with its own set of features enabled. You will need to understand what you are running. For example, let us say you are using Domino for mail with Notes clients only. If nothing changes (same number of users or messages) but you do not manage mailbox sizes, you can expect a chart showing your CPU per active user to creep up over time. As user mailboxes get larger and larger over time, they become more costly for Domino to open, manage, and update.

Since Domino is such a dynamic workload, do not go chasing after every peak and valley. If the issue is not re-creatable and not impacting your server, then spending your time to identify what happen may not worth the effort.

If you have not been tracking your servers, how can you interpret the data when you get into a situation? Many times, the most obvious indicator of what is wrong is to look at what has changed. If you are not monitoring your servers, then you cannot measure them. If you do not measure your servers, then you cannot tune them. As Figure 12-38 shows, how can you tell if the change was good or bad without a complete picture of both the Domino workload and system resources?

13

# Applications

In this chapter, we discuss different methods of accessing Domino data, and then look at some external programs that you can use to access Domino data.

## 13.1 Web servers

There are several different ways to serve HTTP pages from a zSeries system. Depending on your application, you could use the Domino HTTP stack, WebSphere Application Server, or the IBM HTTP server (this is the Apache server when you run Linux on zSeries).

### 13.1.1 Domino HTTP

Domino includes a fully functional HTTP server. It is started by typing: `Load HTTP` at the Domino console. It can also start automatically when the Domino server starts by adding HTTP to the ServerTasks= line in Notes.ini.

#### Advantages

The Domino HTTP server requires no additional setup or configuration, thus making it easy for you to get started. Domino HTTP is also fully integrated with the Domino Directory, so your users' IDs can be managed from a single location.

#### Disadvantages

The Domino HTTP server does a very good job with Domino data, but is less capable at serving non-Domino data. If your Web application contains little Domino data, one of the other choices might be better

#### When to use

We recommend that you use the Domino HTTP server for servers that are mostly serving Domino applications (for example, a mail server where you are giving users access through iNotes).

### 13.1.2 WebSphere Application Server

The WebSphere® Application Server is a standards-based, Web server deployment platform. It enables you to use Java servlets and JavaServer Pages (JSPs) and provides connections to back-end database systems

For detailed information about integrating this product with Domino, see *Domino and WebSphere Together,* SG24-5955, second edition.

#### Advantages

WebSphere is a powerful server that can serve many kinds of enterprise data, not just Domino data. If you have to serve a great deal of data from different sources, WebSphere is a good choice.

#### Disadvantages

While WebSphere is powerful, it can be difficult to set up and configure, especially for those with little previous WebSphere experience.

#### When to use

We recommend that you use WebSphere on those servers that are going to be serving many different sorts of data (such as Domino and DB2), as well as in organizations that have WebSphere experience to set up and manage the server.

### 13.1.3 IBM HTTP server

The IBM HTTP Server powered by Apache is based on the Apache HTTP Server, which is the most popular server on the Web. The version that comes with Linux on zSeries can be used to replace the Domino HTTP stack.

#### Advantages

Due to its popularity, there are many plug-ins for the Apache server which provide a wide range of functionality, that may be otherwise unavailable. Apache also excels at serving static HTML pages, if your Web site consists of a lot of static data, in addition to the Domino data, Apache might be a good choice.

#### Disadvantages

Apache can be difficult to configure, also management must be done outside of the Domino environment.

#### When to use

If there is a need for some specific function that Apache provides, then using it as a replacement for the Domino HTTP stack makes sense.

## 13.2 Connecting to external data sources using DECS

As of the time of writing, the only connection tool that is shipped with Domino 6.5 for Linux on zSeries is Domino Enterprise Connection Services (DECS), and the only DECS connector available is for DB2 7.1 or higher. Lotus Enterprise Integration (LEI) is not part of the initial release, nor are DECS Connectors for any product other than DB2. The DB2 database must be on the same system as the Domino server, or you must configure a connection to another DB2 database through DB2 code on the Domino system.

In addition to the DECS functionality, you can use a Data Connection Resource (DCR) to connect to external data. DCRs are a new feature in Domino 6 which enable you to access external database data directly from your Notes application without going through the DECS Administration database.

This chapter provides an introduction to building an application using a DCR. For more information, see *Domino Designer 6: A Developer's Handbook*, SG24-6854.

### 13.2.1 Configuring DECS

In order to use a DCR, you need to install Domino Enterprise Connection Services (DECS). During configuration of the Domino server, in the Server Audience Screen, place a checkmark beside DECS. This will install the DECS task and configure it to run automatically.

### 13.2.2 Running DECS

DECS begins running as you bring up the Domino server. The DECS task is controlled by parameters in the notes.ini file. The ServerTasks line in the notes.ini file shows all of the tasks that Domino will run on a continuous basis. The DECS task is included on this line.

If, instead, you want to control the DECS task manually, remove the DECS entry from the ServerTasks line. Then when Domino is started, DECS will not start automatically. You must start DECS manually until you change the notes.ini file to again include the DECS task.

To start the DECS task manually, type the following at your Domino console:

```
LOAD DECS
```

Domino will respond with a task response indicating that the task has started. The first time DECS is started, it will create the DECS Administration Database from the administration template that was included as part of the Domino installation process.

To manually stop the DECS task, type the following at your Domino console:

```
TELL DECS QUIT
```

Domino will respond indicating that the task has shut down.

The only supported data source for DECS connections for Linux on zSeries is DB2. Next we tell you how to connect Domino and DB2.

### 13.2.3 Preparing DB2 to connect with DECS or DCR

The name of the DB2 subsystem we used in building our activities is DB2. The name of the database we used is sample, and the username is db2inst1. The DB2 sample database has several tables that were defined when it was built. We accessed them using DCRs.

To start the DB2 database manager, log in to your Linux guest with your DB2 user ID (for example, db2inst1). Issue the command `start dbm`. For more information about starting the database manager and testing your connection to a database, see 10.5, “DB2: Starting the database manager” on page 269.

Also, ensure that DECS has been started. You can have it start automatically when your Domino server comes up, or you can issue a command to start it. There is a utility called dctest that will test the connectivity of your workstation client through DECS to DB2. For more information about starting DECS and testing connectivity, see 10.6, “DECS: Starting and testing connectivity” on page 270.

### 13.2.4 Connecting Domino with DB2 using DCR

We created a simple Notes database to show how to integrate a Domino application with DB2 databases. It is an organization database that holds employee data such as names, address information, expertise, and salary information. We kept the application simple so we could concentrate on the integration features.

The first thing we did was to create a DCR named Test DCR using the “Shared Resources -- Data Connections” section of the designer; see Figure 13-1 on page 353.

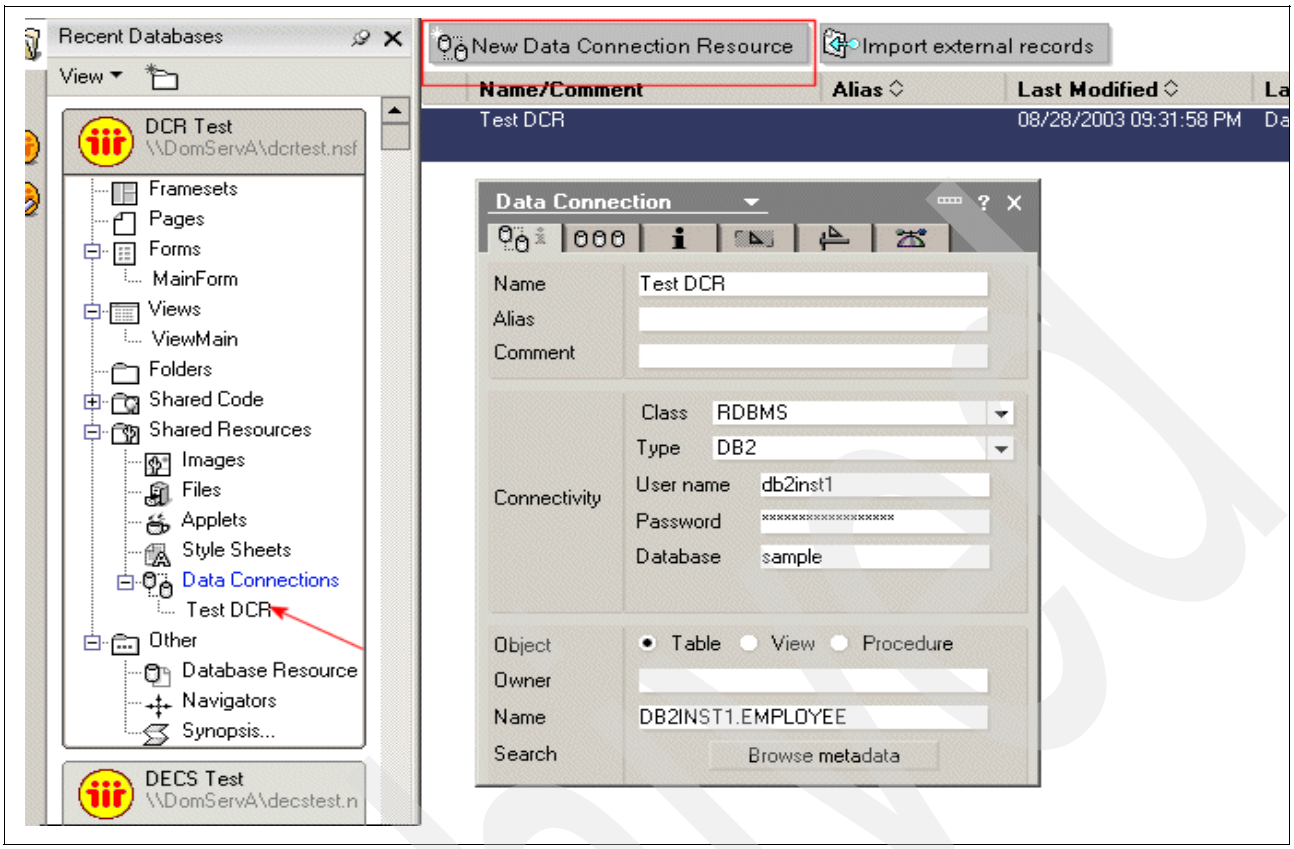



*Figure 13-1 Create DCR*

When creating the DCR, we specified the Class and Type of database to connect to. (This can only be RDBMS and DB2, as of the time of writing.)Then we entered the database name, the username, and password for the user on the DB2 system; see Figure 13-2.

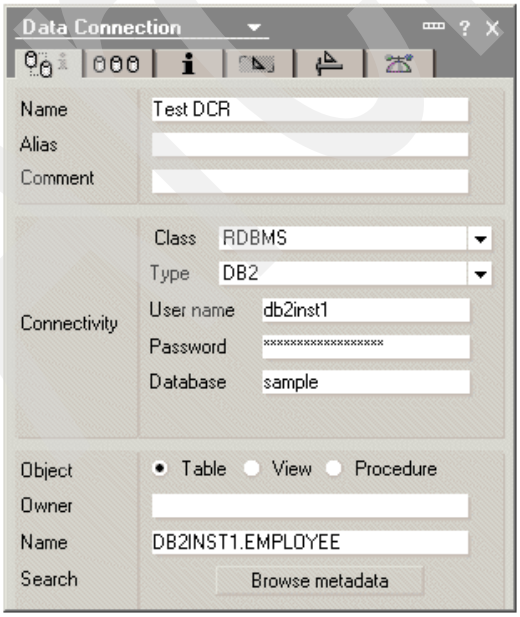



*Figure 13-2 DCR properties*

By pressing the **Browse metadata** button at the bottom of the property box, we could browse the tables in our DB2 database and select the one that we needed need to use for our connection; see Figure 13-3.

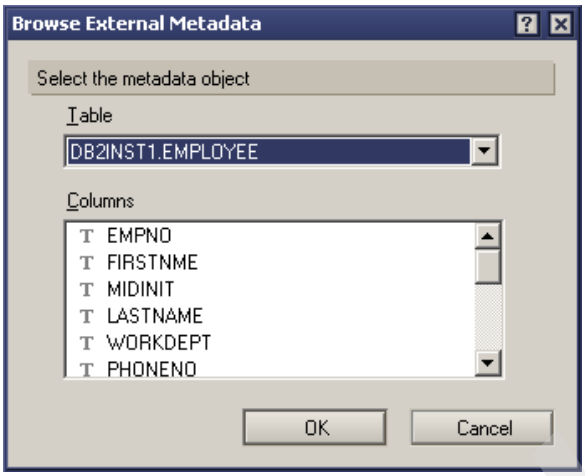



*Figure 13-3 Browsing external metadata*

Next we created a form with fields that we mapped to the fields in the DB2 database. When we created a field, we checked the “External data source” checkbox in the properties box and the data source portion of the dialog box opened; see Figure 13-4.

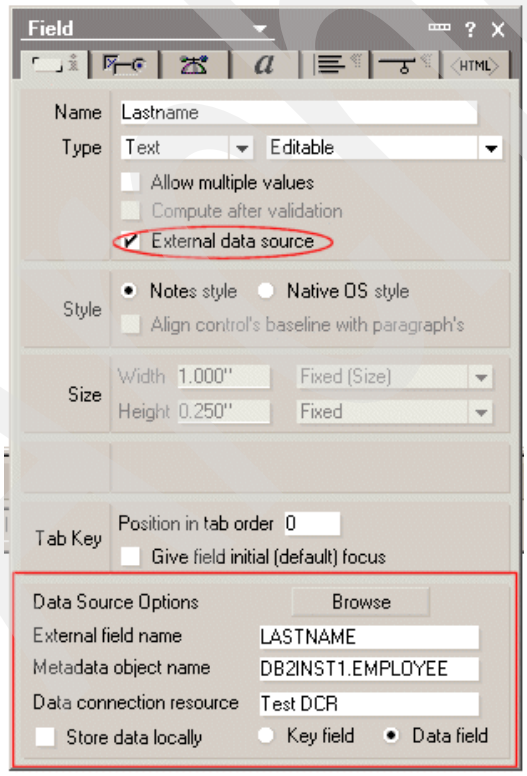



*Figure 13-4 DCR field*

Pressing **browse** allowed us to select the DCR, table, and DB2 field that we wanted mapped to our Domino field; see Figure 13-5. Note that the type of field (text, numeric, date, and so on) is indicated by the icon to the left of the field name.

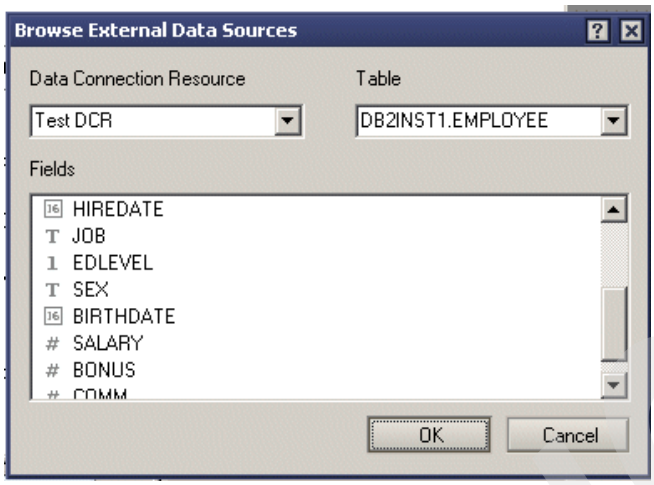



*Figure 13-5 Browsing the DB2 data*

When we set up the mapping, we also must choose whether the field is a key field or a data field. If it is a data field, we must also choose if the data is to be stored locally.

Key fields are the link between the form and the backend database or application. Note that key fields are always stored locally, as well as on the backend.

Storing the data locally means that the data is always available to Domino. In Domino language, this is the equivalent of turning a “computed for display” field into a “computed” field.

After we defined our DCR and fields, we had to import the key fields into our Domino database from DB2. We clicked **Import external records** in the action bar of the designer client; see Figure 13-6. This creates “stub” documents in the Domino DB with just the key fields in them. Then, when the document is opened, the real time portion of DECS populates the data from the DB2 database.

After we imported the key fields, we had to check the “Allow connections using DCRs” box in the database properties box. Importing keys will not work if the checkbox is selected; you will receive an error message.


*Figure 13-6 Importing key database fields into Domino*

Figure 13-7 on page 356 shows the form that stores the employee data. All the fields are text, except for:

- DispName:, which is a computed for display field that proper cases the person’s name
- Birthdate:, which is a date/time field

- Age:, which is a computed field that calculates the person's age based on their birth date

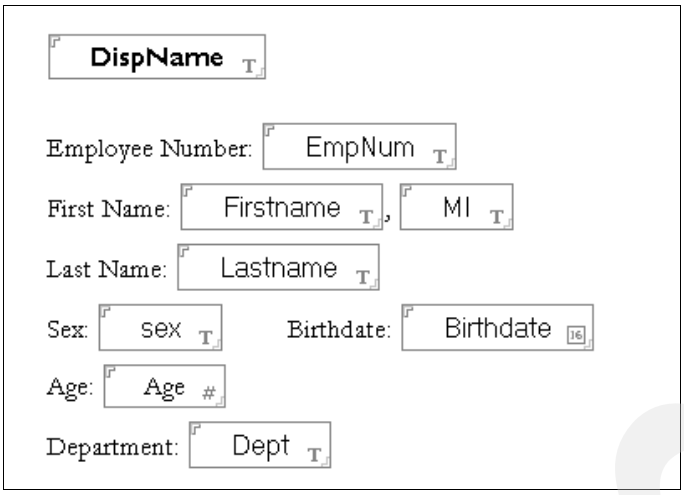



*Figure 13-7 Form design*

When the document is opened, DECS pulls the data from the DB2 database and populates the form. Figure 13-8 shows the form with live DB2 data displayed.

The Notes client can then be used to update and maintain the DB2 database, and Domino replication can be used to distribute the data throughout an organization.

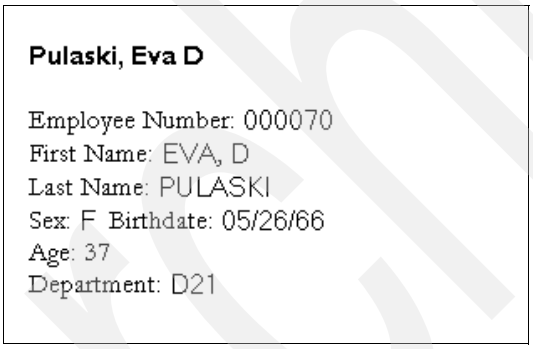



*Figure 13-8 Live DB2 data*

This gives you a brief overview of the use and power of DCRs. For more information, refer to these IBM Redbooks:

- *Domino Designer 6: A Developer's Handbook*, SG24-6854 for more information about DCRs
- *Lotus Domino for S/390 Release 5: Enterprise Integration Using Domino Connectors,* SG24-5862 for more information about DECS and zSeries

14

# Migrating to Domino 6.5 for Linux on zSeries

In this chapter, we discuss how to migrate your Domino applications and mail files from other server platforms onto Domino for Linux on zSeries. We show techniques you can use to move your data, and point out some of the pitfalls that can occur when moving between various platforms. We do not cover all the new features of Domino 6, but you can find this information in *Upgrading to Lotus Notes and Domino 6*, SG24-6889.

## 14.1 New or special in Domino 6.5

In the Notes Designer, there are few changes in Domino 6.5 compared to Domino 6. This holds true for the other clients as well, so there should be no need for you to change any Domino 6 applications when moving to Domino 6.5. Refer to 1.5, “What’s new in Domino 6.5” on page 11, and also to the Release notes for Domino 6.5 and the Notes Designer help, for more information.

Perhaps the most important change in Domino 6.5 for a Notes developer is that the template for Domino Web Access (DWA, formerly known as iNotes) is customizable. Using Domino Designer, developers can add action buttons to views or dialog boxes, provide additional choices for the Welcome Page, and replace the Domino Web Access logo with a corporate logo.

## 14.2 Upgrading to Domino 6.5

If you upgrade from Domino R4 or R5, then we recommend that you read the designated IBM Redbooks, such as *Upgrading to Lotus Notes and Domino 6,* SG24-6889, concerning upgrading to Domino 6. In addition, we highlight some important points for your upgrade project in the following sections.

Figure 14-1 on page 359 shows our recommended upgrade strategy. Begin with the administration clients, then check all your customized templates, and upgrade your Domino Directory. Next you should upgrade the administration and hub servers. When all those servers are stable, you can upgrade your remaining servers. Finally, upgrade the Notes clients.

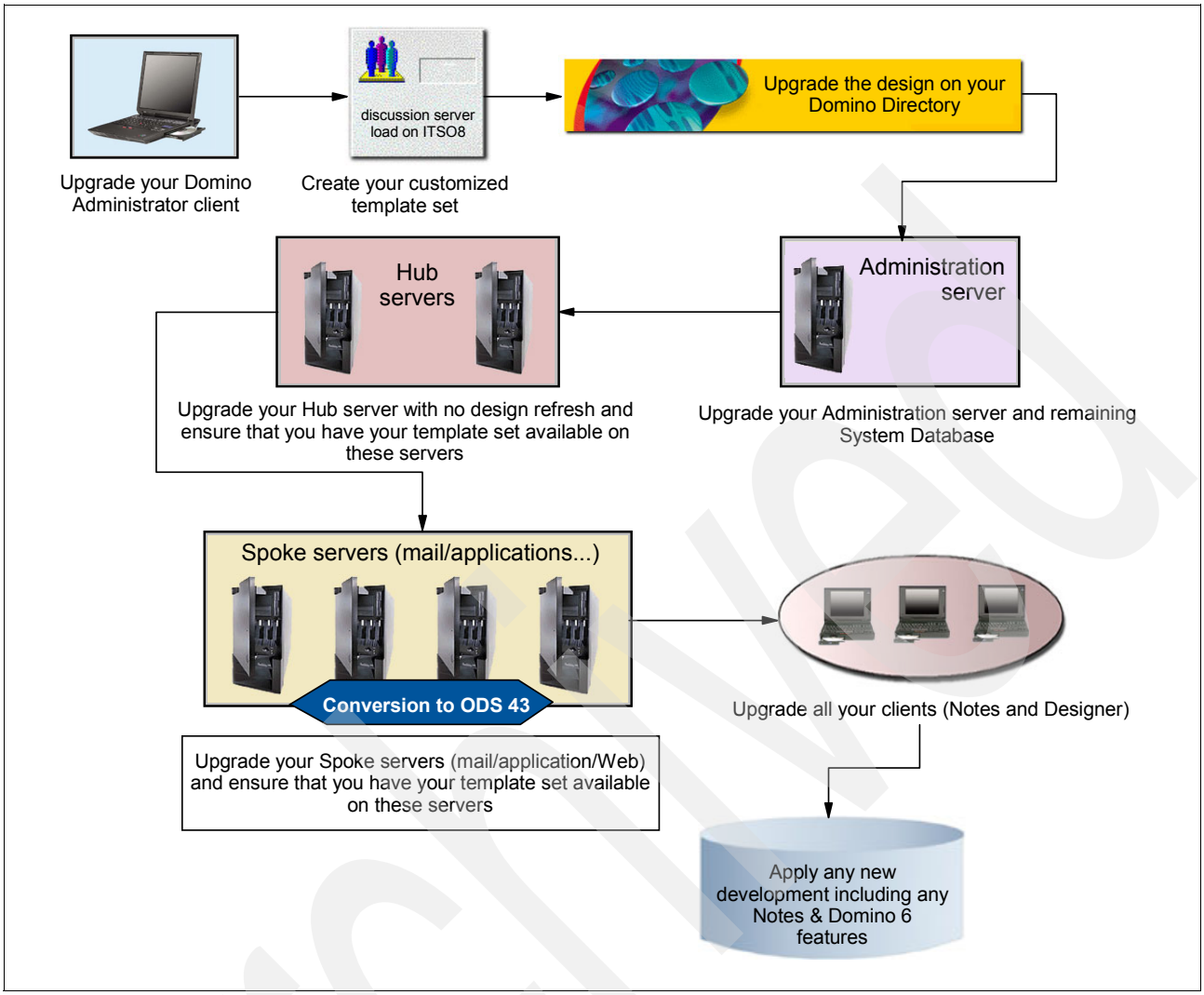



*Figure 14-1 Upgrade strategy to migrate to Domino 6*

If you start from Domino 6.0.x, there should not be much to do. We suggest that you test your business-critical applications and your customized Lotus standard templates, such as pubnames.ntf. After successful testing, you can apply the new Domino 6.5.x version to your servers.

> **Note:** We strongly recommend that you do *not* change or customize any of the standard templates!

In general, all non-customized standard Lotus templates will work. That means you can just upgrade these databases to the new design. Also, your applications that only use “standard” LotusScript and @functions will work. However, you should be aware that the implementation or behavior of Lotus connectors such as LSX and DECS, or the use of the C/C++ toolkit, may be different from earlier Domino releases.

In addition to upgrading the Domino version, you also need to plan for a platform change. The easiest case is when you come from another UNIX flavor. Migrating from a Windows platform is slightly different from that. We discuss both scenarios in the next sections.

### 14.2.1 Steps from other UNIX flavors (AIX, Solaris, z/OS USS)

As mentioned before, it is very easy to migrate from one UNIX platform to another. You do not have to worry about code pages, case sensitivity, and new filesystems. For most of your Domino servers and databases, a simple transfer to the new platform should be sufficient. But before planning or doing a migration, verify that the prerequisites of your own applications are met by the desired platform. Not all Lotus products or third party products that your applications require might be available.

There is one exception to this. When migrating from Domino on zSeries from z/OS to Linux, there are changes to the way the filesystem is set up (DASD instead of HFS or zFS datasets), and of course the character set is a different one for z/OS and Linux (EBCDIC versus ASCII).

### 14.2.2 Steps from Wintel

Changing from a Windows platform to Linux is slightly more complicated than changing from UNIX to Linux. We discuss the high level issues here, and offer more detailed information in 4.3, “Brief introduction to Linux and UNIX filesystem” on page 57.

#### UNIX is case sensitive

In Windows, filenames are not case sensitive—but in UNIX, they are. If your scripts call for the file log.nsf and the file is listed as LOG.NSF at the operating system level, the file will not be found when the script runs. We recommend that you name every file only using lower case. In this way it will be easier to find misspelled filenames.

#### Backslash

The backslash (\) character, used to separate directories under windows, is not present in UNIX. Just as with Web addresses, use the forward slash (/) between directories.

In the Domino Directory, the Domino server uses the backslash \ for path names, and converts it when accessing the UNIX filesystem. However, it may be a good idea to use the forward slash / wherever possible. This automatic translation does not work for scripts, agents, and so on (so make sure they use the correct delimiter). But it does work for the Notes client, so the user can still use the backslash \, such as in the open database dialog.

#### Code pages

Be aware of differences in code pages for Wintel and Linux. Linux is an ASCII-based operating system, so there normally is no need for a special code page translation, such as that required for exchange text files between z/OS and the rest of the world.

However, in some cases Windows and Linux may have different interpretations of special characters, so you should double-check critical files to make sure they look and work the way you want them to. Those differences do not affect text in normal Notes databases, but they might cause problems in detached files on the operating system level, as well as interpretations for Web sites (http task).

### 14.2.3 Server consolidation tips and methods

#### Plan for DASD

Chapter 4, “Disk configuration” on page 53, contains a detailed explanation of how to design the filesystem and mountpoints for Domino for Linux on zSeries. In this section, we offer a just a brief summary of tips and methods.

We recommend that you spread the databases equally over the data directories and servers. “Equally” in this context means trying to balance your database distribution in terms of size, grow rate, IO activity, workload, and other factors.

**Tip:** First - plan your new Domino infrastructure. And remember to plan for servers for testing and application development.

## Migration scenarios

There are two scenarios for migrating your Domino infrastructure to Domino on Linux for zSeries:

- Scenario 1 - with server consolidation (fewer Domino servers on Linux for zSeries)
- Scenario 2 - without server consolidation (the same number of Domino servers as before)

In the following discussion we focus on scenario 1, because scenario 2 can be treated as a special case of scenario 1 with one exception: you can take the IP address and the server ID from your old server and move it to the new server (this way you do not have to worry about informing your users or the Notes client about a new server). Which scenario you choose depends mainly on your existing Domino infrastructure.

When you plan your new environment, it is a good time to rethink the way you use your Domino servers. We recommend that you separate your servers by function.

For example, have a dedicated server for Notes mail, for POP3 and SMTP mail, for Notes applications, for Web applications, and one admin server, as shown in Figure 14-2. If possible, do not mix these kinds of workloads on one Domino server; it is easier to solve problems, maintain updates, and tune and administer servers that have dedicated workloads than servers with a mixed workload.

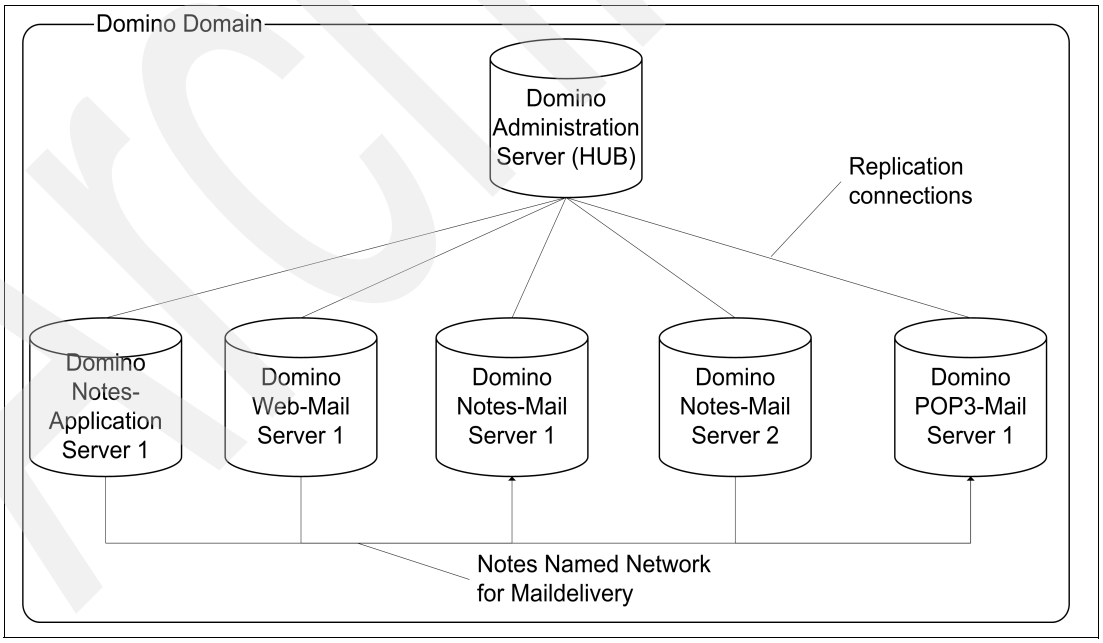



*Figure 14-2 Ideal Domino Domain*

### 14.2.4 Migrating techniques: Domino replication versus FTP

Migrating from one Domino platform to another means moving a great deal of data. Most of this data will be contained in Domino databases. So the obvious method is to use replication as the preferred transport mechanism. A special sort of replication that can be very useful for migration is provided by a Domino cluster. But FTP is also a good, standardized, reliable data transportation method. In this section we discuss the pros and cons of both methods, and help you to decide when to use which method.

#### Using replication

When using replication as a transport mechanism between your old and new Domino servers, you have to prepare some things before you can start. You have also decide whether you want to use the normal replication, or whether you want to set up a cluster for the migration and data transfer process. Here we list some of the tasks you have to perform in setting up the replication, and also mention special items for clustering.

First, you have to create the replica stubs on the new server for all the databases you want to move. Next, you should create a connection document pointing from your old Domino server to your new Domino server. Enable the replication and check the Domino log for possible replication errors. Finally, inform the users and their Notes clients about the new database location.

Here are some of the difficulties you might experience when using replication:

- Private folders and views in databases will not replicate.
- Some field settings for documents (author, reader fields, ...) can prevent the Domino server from replicating these documents.
- ACL and replication settings for databases configured by a user may also prohibit the server from replicating.
- If local encryption is enabled for a specific user ID, the server is not able to read, and therefore replicate, the database.

Keep these points in mind when using cluster replication:

- Make sure that the “Server to run on” property in any agents is changed to the “new” server before you shut down the old one.
- Are your applications cluster-enabled? What happens to data on non-cluster-enabled databases when clustered?
- If your source server is overloaded, it may not be able to handle the additional workload generated by the cluster.
- The maximum recommended number for servers in a cluster is six. Attempting to consolidate more than five “other” servers at one time onto your zSeries server can cause problems.

The great benefit of using clustering is that you do not have to worry about informing your users or their clients about the new server; that is automatically taken care by the cluster process. As a result, migrating is simple: as soon as the cluster is enabled and running for all the databases, wait until all clients have the cluster information. Then you can shut down the source server and the failover process will direct the clients and users to the new server.

#### Using FTP for transferring your databases

FTP, on the other hand, is a fast transfer method that does not look inside the data. So FTP does not care about ACL, replication, or encryption settings. On the down side, normally you

have to shut down your Domino server to use FTP, so you only can do this during your maintenance window.

**Note:** There are two modes of file transfer in FTP - binary and ASCII:

- Binary transfers are an exact copy, and no reformatting of the file is done by FTP.
- ASCII transfer assumes the file you are transferring is a text file and, when transferring between platforms, will attempt to reformat the file to the native text format of the destination machine.

If you are in ASCII mode when transferring a database, the database will be unreadable by Domino on the destination machine. Some versions of FTP start in ASCII mode. Therefore, you should always type: **bin** on the FTP command line to ensure that you are in binary mode before transferring any databases or templates.

If you using FTP as your transfer method, make sure that both the target and the source Domino servers are shut down. If you cannot shut down your target server, make sure you FTP the databases with different filenames.

As shown in Example 14-1, the local file log.nsf is stored as log.nsf.new on the remote system. (Be sure to rename the file after the transfer is complete.)

*Example 14-1 Using different filenames during FTP*

```
C:\Lotus\Notes\Data>ftp linuxa
Connected to linuxa
220 ready
User (linuxa:(none)): balu
331 Please specify the password.
Password:
230 Login successful
ftp> cd /domserva/notesdata
250 Directory successfully changed.
ftp> bin
200 Binary mode selected.
ftp> put log.nsf log.nsf.new
200 PORT command successful.
150 Go ahead send the data.
226 File receive OK.
ftp: 1064448 bytes sent in 0.17Seconds 6261.46Kbytes/sec.
ftp> bye
221 Goodbye.
```

Here is how to rename the file after the transfer:

```
linuxa:/domserva/notesdata # mv log.nsf.new log.nsf
```

This procedure is very important when you ftp into a transaction-logged server. Otherwise, the server would start transaction logging during FTP process, which could corrupt the transaction log.

After you finish the file transfer, check the setting of the file permission bits. In 4.3.6, “File permissions” on page 58, we list the typical permission bit settings for Domino servers. As mentioned, FTP does not care about ACL setting—but your target Domino server does. So make sure the ACL settings allow the new server to access the databases you transfer.

### 14.2.5 Migrating scenarios

> **Note:** Many of the tasks we list here can be performed by the Domino server itself, using the adminp server task. However, success will depend not only on a well-configured adminp task, but also on the database template (special mail template) you use.
>
> We discovered in some situations that “server-controlled” migration can fail if you are not using the original Lotus templates. With that in mind, we describe the migration steps in extreme detail.

There are two scenarios for migrating Domino servers: run the new and old Domino servers concurrently for a certain time, or replace the existing old server with the new Domino server. In the following section, we explain each scenario in detail.

***To run the new and old Domino servers concurrently for a certain time***

Follow these steps:

1. Install and configure the new Domino server.
2. Copy the databases to the new Server.
3. Create a connection document for replication between the old and new Server and start the replication.

**This step can be automated!**

4. Inform your users about the new server and location of the databases. Make sure the Notes client gets updated with the new server and database information (Mailservers).
5. Shut down the old server after the expiration of the time limit.

***To replace the existing old server with the new Domino server***

Follow these steps:

1. Install the new server.
2. Copy the key datasets from the old server to the new one (notes.ini, cert.id, server.id, names.nsf, …). Customize notes.ini for directory entries.
3. Copy the Domino databases to the new server.
4. Switch the IP addresses from the old server to the new server.
5. Shut down the old server and start the new Domino server.

Both scenarios are valid; the one to choose depends on your individual situation. However, we recommend the first scenario because it is more fault-tolerant and there is a built-in fallback if things go wrong. Another advantage is that you can make your migration more granular and slowly ramp up the load on the new Domino server.

For migration and server consolidation, you might consider a contract with a services vendor. IBM Global Services and Tiassa Technologies have Domino migration experience[1], as do other vendors who might be available in your area.

## 14.3 Moving your applications from Windows to Linux

If your applications function today on Domino server running Windows NT or Windows 2000, they will also work on a Domino for Linux server.

[1] IBM Global Services and Tiassa Technologies provided authors for this redbook.

Domino databases are platform-independent, meaning that you can copy files from NT to UNIX and open the database without any kind of change to the file format. However, there are a few considerations to bear in mind due to the differences in the environment.

The Decommission Server Analysis Tool gives you detailed information about the connections, databases, and other items two servers have in common. It can be used as a starting point for migrating your servers to your zSeries system. The Decommission Server Analysis Tool can be found in the admin client, on the Server>Analysis tab, under the Analyze tool. This tool allows you to do the following:

- Create an inventory of your existing Domino applications and include such items as replicas (on which servers), database properties, clustered, ACL settings, size, path, last used, inherits design from, is template.
- Find out who is responsible for the application, and if it is still needed.
- Find out how important the application is for your organization and how much workload it will generate on the server. This information is important for deciding when, and onto which server, to migrate the application.
- Archive and delete old and unused applications.
- Analyze the remaining applications for possible incompatibilities, such as dependencies on third party products, external data source as DB2 or Oracle, or specific hardware (such as a fax modem). Check the application using the design synopsis feature of the Notes Designer client or other designer (for example, Teamstudio) and appropriate filter rules (R4-R6, R5-6).
- Make the appropriate changes to the design of the database.
- Test the application on the new platform.
- Plan the distribution of your applications (which server, which directory, cluster).
- Transfer the databases to the target server.
- Notify the users and the Notes clients of the new database location.
- Monitor the migration process

## 14.3.1 Key migration questions

To ensure that your application will be compatible, consider the following questions before moving an application from Windows NT or Windows 2000 to Linux.

- Is your Domino application “self-contained”?
- Did you use Computer Aided Software Engineering (CASE) tools?
- Does your application use OS platform exploitation?

### Is your Domino application “self-contained”?

A self-contained application runs entirely inside the Domino server, without any explicit references to files, without external calls, and without importing or exporting data. An explicit reference to a file, such as c:\domino\data\NAMES.NSF, will not work on Linux and needs to be replaced with /domino/data/names.nsf.

Linux does not support the \ character for specifying paths, and uses the / character. Linux is case sensitive when specifying paths and filenames, while NT is not. Case sensitivity can also be a problem anywhere an external script call, link, or hot spot is used; be sure to verify that the correct case is used.

### Did you use CASE tools?

While CASE tools may be helpful, many of these tools were created with non-UNIX operating systems in mind, and their output code may not be compatible with Linux. Be sure to check with the manufacturer for compatibility before using these tools.

### Does your application use OS platform exploitation?

Anything in the application that might be platform-specific could fail in the Linux environment. NT-specific services, NT Registry Sync for user registrations, Active-X controls, or compilers that rely on platform-specific libraries to compile the application will cause problems when the application is moved to Linux.

## 14.3.2 Moving the application to the Linux server

Transferring the files from Windows NT or Windows 2000 to UNIX can be done using many methods: FTP, transfer via CDRW, Iomega Jazz drives, or other media and PCNFS can all be used. For this example we used FTP, since it is the most common tool used in the field.

Since FTP servers are installed by default on the UNIX side and not on the NT side, it is usually easier to open an FTP session from the NT box and connect to the UNIX box. This example uses NT 4.0:

1. From the NT box, open an MSDOS command prompt by selecting **Start** -> **Programs** -> **MSDOS**.
   a. Change directory to the server's data directory:
      ```
      cd \lotus\notes\data
      ```
   b. Start an FTP session:
      ```
      ftp servername
      ```
2. Change directory on the UNIX box to the data directory:
   ```
   cd /local/notesdata
   ```
3. Switch to binary transfer mode:
   ```
   bin
   ```
4. Transfer the databases:
   ```
   put names.nsf
   ```
   Or, transfer multiple files at once using wildcards with the **mput** command:
   ```
   mput *.nsf
   ```

> **Important:** Never add or remove databases from the OS level while the Domino server is up and running. Domino caches the data directory listing, and unpredictable behavior can occur if you modify the data directory while the server is running. This could result in a server crash or hang.

### Ensuring permissions are correct

After the transfer is complete, make certain that permissions are correct on the UNIX machine. Log in to the UNIX machine and change to the data directory (cd /local/notesdata) and check the permissions on the transferred file:

```
ls -l *.nsf
```

Following is an example of the permissions line:

```
-rwxrwxrwx  1 nadmsup notes  1589248 Feb 22 09:34 log.nsf
```

Interpret this record as follows:

- The first column shows the permissions. The leftmost letter indicates whether this is a file or a directory. A dash (-) in the left position indicates it is a file; a letter d in the left position indicates it is a directory.

  The next nine letters indicate the access rights to the file for the owner, group, and world, given in 3-character segments.

  From left to right, the permissions in each segment are read access, write access, and execute access. Therefore, an entry of rwx means that read, write, and execute access is granted. If any of the letters has a dash (-) in its place, then that permission is not allowed. For example, r-x means that read and execute access is given, but write access is not.
- The owner is the user ID that owns the file, which is indicated by the third column in ls -l. In this case, it is “nadmsup.” The owner’s permissions are read from the first three permission characters in column 1 (following the file or directory indicator).
- The group is identified in the fourth column. In this case, it is the “notes” group. The group’s permissions are identified in the next three characters in column 1.
- The world is anyone else who has login access to this system. Their permissions are specified in the last three characters of column 1.

Since the Domino server is the only one that should be changing or directly reading the databases, and databases are not executable programs, the permissions for databases should be:

```
-rw---- 1 nadmsup notes 1589248 Feb 22 09:34 log.nsf
```

If the permissions are not correct, you can issue the command:

```
chmod 600 filename
```

Where, filename is the name of the file on which you want to change the permissions. This will give read and write access to the database for the Notes user, but will not allow anyone else to view it. Since the Domino server runs under the Notes user account and makes all of the read and write calls on behalf of the clients, most organizations will want to keep the access to the files restricted to the Notes user account.

### Checking for case sensitivity

In NT, filenames are not case sensitive, but in UNIX they are. If your scripts call for the file log.nsf and the file is listed as LOG.NSF at the OS, the file will not be found when the script runs. After the FTP completes, check to ensure that the filenames are in lowercase unless your application is specifying otherwise.

```
ls -l
-rw---- 1 nadmsup notes 1589248 Feb 22 09:34 LOG.NSF
mv LOG.NSF log.nsf
ls -l
-rw---- 1 nadmsup notes 1589248 Feb 22 09:34 log.nsf
```

**Note:** You should always type: **bin** on the FTP command line to ensure that you are in binary mode before transferring any databases or templates.

## 14.3.3 Other information for migrating Domino applications

In the following sections, we describe changes that have been made in Domino 6 for which you should check your applications.

### The default fonts in Notes 6 have changed

In Notes 6, the default fonts have changed. If you have developed your current applications using the default fonts in Release 5 and earlier releases, you should check for screen corruption in the display of various parts of your application with Notes 6.

### Formula language has been rewritten in Notes 6

In order to improve performance and maintainability, Notes Formula language has been rewritten in Notes 6. Emphasis has been placed on making sure that the new code provides 100 per cent compatibility. Therefore, no application changes are required. While performing migration testing, however, be sure to exercise all applications that use Formula language.

### MQSeries® Lotus Script Extension (MQLSX) incompatible with Domino 6

MQLSX is not supported (although that does not mean MQLSX does not work) with Domino 6. It is also unsupported by Lotus after December 31, 2004. You should carefully test all applications that use MQLSX. We recommend that you rework those applications which use MQLSX. Developers should modify their applications to convert LotusScript to a Java API prior to their application servers being upgraded to Domino 6.

### Strict Date/Time field interpretation has been restored in Notes 6

In Release 5, “robust” interpretation of date/time fields was introduced. Prior to Release 5, if an end user entered an incomplete or invalid date in a date/time field, they would be presented with an appropriate error message and would need to correct the date.

In Release 5, this capability was changed so that if an end user entered an incomplete or invalid date, Notes would make an educated guess as to what the date should be and not present the user with an error message. However, the resulting date could be one that the end user did not intend. For example, if you enter “24/06/2001” in a date field and press TAB, Notes guesses that you mean “Sat 06/24/2001” and changes it as such.

In Notes 6, the behavior that was prevalent prior to Release 5 is restored as the default behavior. Users may change to the Release 5 behavior by unchecking the new User Preference “Strict Date/Time Input”. This preference may be changed in User Preferences on the Basic tab in “Advanced Options”. There is no need to change applications. However, your users may experience differences on how date or time fields are evaluated, depending on what client they are using. You may want to consider additional documentation or validation processes to make sure that your users have entered the dates they intended.

### Some JavaScript events have been changed in Notes 6

The implementation of JavaScript events that are common to Notes clients and Web clients has changed in Notes 6. In Release 5, support for JavaScript events was introduced in the Domino Designer. For Web applications, this meant that you could easily specify the code required for any JavaScript event. A subset of JavaScript events were also implemented for the Notes client. For events that were common to both the Notes and Web client, you could specify code once in the event, and it would work the same in both environments.

In Lotus Domino Designer 6, you can now specify separate behaviors for the common events, dependent on where it will execute (Notes client or Web). This allows applications to tailor their capabilities to the client that is being used, without having to use conditional code in the event logic.

The Notes client version of the event will now let you use LotusScript or Formula Language, in addition to JavaScript. One key question is how do you migrate designs between R5 and

Notes 6. In addition, Notes client events that are the equivalents for JavaScript events are being “deprecated”. This means that application programs should eventually move code in the deprecated Notes Client event to the JavaScript equivalent.

The Notes Client events that are being deprecated and their equivalent JavaScript events are listed in Table 14-1.

*Table 14-1 JavaScript events*

| “Deprecated” Notes Client event | Equivalent JavaScript event |
|---|---|
| Post Open | onLoad |
| Field Entering | onFocus |
| Field Exiting | onBlur |
| Click | onClick |
| QuerySave | onSubmitt |
| QueryClose | onUnload |

In Domino Designer Help, you find the following descriptions:

Forward Compatibility:
R5 applications running in the Notes 6 client behave the same. Those JavaScript events that worked in the Notes client continue to work on both the Web and the client. The deprecated events continue to work. When an R5 application is saved in Domino Designer 6, JavaScript events are moved into their corresponding Web and (where applicable) Client events. The deprecated events are also moved. For example, the onBlur event is moved into the onBlur - Web and onBlur - Client events, and the Exiting event is moved into the Exiting (deprecated) event.

Backwards Compatibility:
Notes 6 applications running in the R5 client behave the same as in the Notes 6 client as long as they are not recompiled in R5. Those events designed to run on the client run on the client and those events designed to run on the Web do not run on the client. The deprecated events work. If a Notes 6 application is recompiled in R5 Domino Designer, Web events are moved into their corresponding JavaScript events but client events are lost. The deprecated events are moved.

We suggest that developers take the following actions as appropriate:

Forward Compatibility:
For the JavaScript events that run on both the Notes client and the Web, where you use conditional code to modify the behavior for one environment or the other: instead use two routines, one for the client and one for the Web. For the deprecated events: move the code to the corresponding “on” client event.

Backward Compatibility:
If you continue to design in both R5 and Notes 6, you should keep the deprecated events and use the same code in the client “on” events as in the Web event.

### @Command([ToolsUserLogoff]) and password prompt

@Command([ToolsUserLogoff]) can produce a different user experience when used to force password prompt. In R5, some applications use the @Command([ToolsUserLogoff]) to force the user into a password prompt situation.

This scenario might be applicable in cases where an application wants the user to specify a password before submitting a form. The application would issue the @Command([ToolsUserLogoff) command followed by some other processing (such as document Save) which would force the password prompt dialog to be displayed to the user.

In Notes 6, there is new user preference called “Clear screen on Logout”. If an user has this preference enabled and the @Command([ToolsUserLogoff]) is issued followed by processing such as a document Save, the user's screen is cleared and the password dialog is displayed.

In R5 (or if the preference is disabled), only the password dialog is displayed. The application still works correctly. You do not need to change your applications, but you might consider additional user documentation or perhaps a different implementation that does not involve @Command([ToolsUserLogoff]).

More information about the changes between R5 and Domino 6 can be found in Domino Designer Help and at the following Web site:

http://www.lotus.com/ldd/today.nsf/lookup/designer_rnext_technical_overview

**Note:** Check for filesystem references, servernames, and so on in your applications and customized templates

In Table 14-2, we list commands and @functions that may cause problems during a migration from R4 to Domino 6. Some of these commands should also be double-checked if you migrate from Wintel to an UNIX platform.

*Table 14-2 @functions and commands to check for*

| | File handling | Shell calls | External Data access | External Programs | Date Time format |
|---|---|---|---|---|---|
| **Lotus script** | Access, Append, ChDir, ChDrive, Close, CurDir, CurDir$, CurDrive, CurDrive$, Dir, Dir$, EOF, FileAttr, FileCopy, FileDateTime, FileLen, FreeFile, Get, GetFileAttr, Input, Input$, InputB, InputB$, InputBP, InputBP$, Kill, Loc, LOF, MkDir, Name, Open, Output, Put, Random, Read, Reset, RmDir, Seek, SetFileAttr, Shared, Width, Write | ActivateApp, Command, Command$, Execute, Shell, Yield | Environ, Environ$, GetThreadInfo, IMESetMode, IMEStatus, Line, LMBCS, SendKeys | Alias, Any, Declare, Lib, UseLSX | Date$, DateValue, Format, Format$, Time, Time$ |
| **@ Function** | AttachmentDetachAll, AttachmentLaunch, AttachmentProperties, AttachmentView, EditInsertFileAttachment, EditInsertObject, FileExport, FileImport | Execute | @DDEExecuted, @DDEInitiate, @DDEPoke, @Environment, @SetEnvironment | @DBCommand | |

## 14.3.4 Migration considerations for mailservers

The migration of an existing messaging infrastructure from multiple Domino servers to a central Domino platform presents a project team with a number of challenging tasks. A stable

system platform with all proper disciplines must be provided, and the successful migration on the Domino level itself must be accomplished. IBM Global Services (IGS) can provide tools and assistance during migrations such as these.

The following tasks should be performed for a successful mail migration:

- Identify the users you want to migrate in this step by one or more of the following criteria:
  - Organizational affiliation
  - Home- or mail server
  - Individual selection
  - Mailfile sizes
- Generate a balanced distribution of mail databases on the target Domino server according to several criteria:
  - Maximum size of a directory
  - Number of databases per directory
  - Number of directories per server
  - Number of servers
- Transfer the databases using FTP or Domino replication:
  - Assure of a reliable data transfer
  - Log progress of the data transfer

- Adapt the Domino Directory
  - Convert the relevant person documents
  - Convert the relevant mailin-database documents
- Adapt the essential parameters in the client's location documents
  - Select the relevant location documents by type or name
  - Adjust calendar entries if daylight saving time (DST) settings have changed
  - Adjust the DST parameter

    **Note:** DST might be an issue, especially when you migrate from R4.
  - Adjust the agent "ProcessOutOfOffice" to the new server
  - Add the new icon to the users workspace, or add the new bookmark list entry
  - Log the progress of the migration
- A migration is a good time to clean up your mail environment, in terms of:
  - Standardizing naming conventions on a filename level
  - Reassessing reference inconsistencies in the Domino Directory
  - Monitoring the size-distribution of all mailfiles
  - Imposing Quotas for the mail databases
  - Defining meaningful mail server groups and cleaning up access groups
  - Setting the user access to Editor (the default Domino 6.5 ACL setting for mail databases)

Figure 14-3 on page 373 summarizes the basic tasks during a migration or server consolidation:

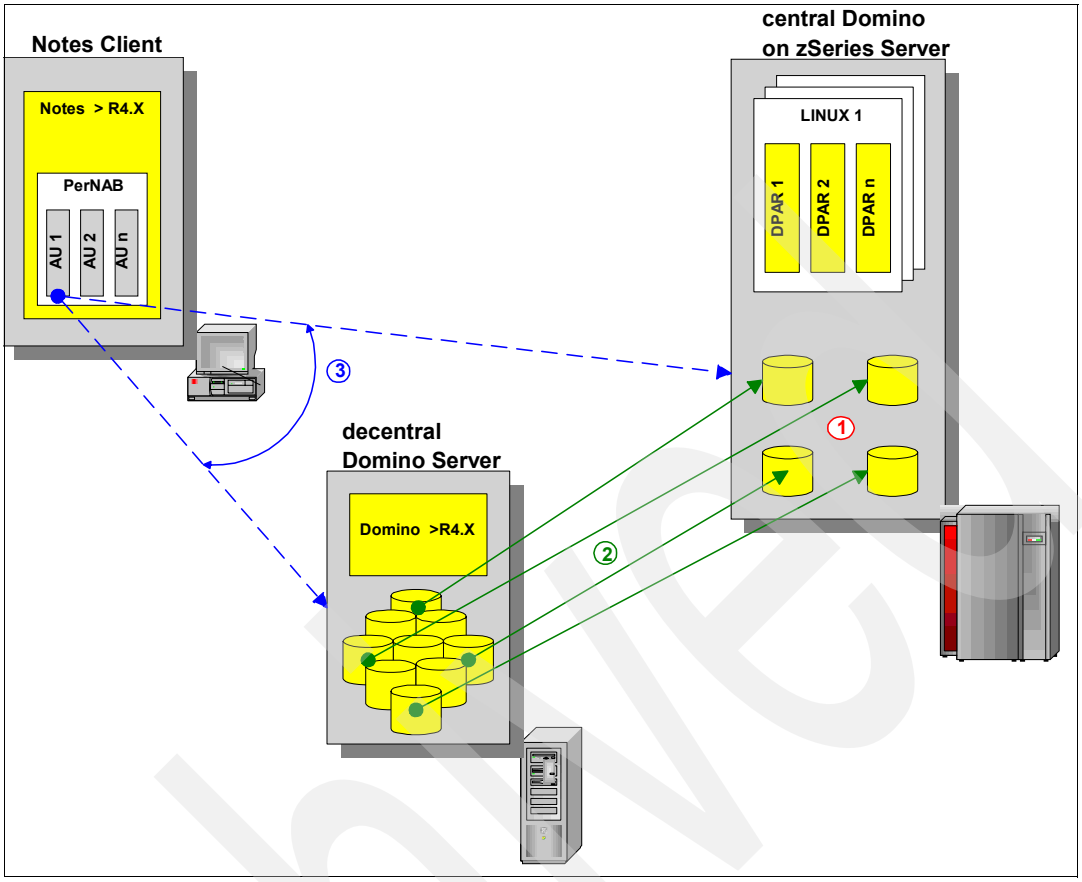



*Figure 14-3 Migration steps*

The steps shown in Figure 14-3 indicate the following:

1. Plan for a well-balanced distribution of the databases (server and directory), and define the filesystem on your new Domino server.
2. Transfer the databases from the old servers to the new Domino server.
3. Redirect the Notes Client to the new server. Also customize other client settings to meet the requirements for the new Domino server version and platform.

A

# Our test system

In this appendix, we describe the system we used during the project for this redbook. Figure A-1 provides a graphical overview of the system.

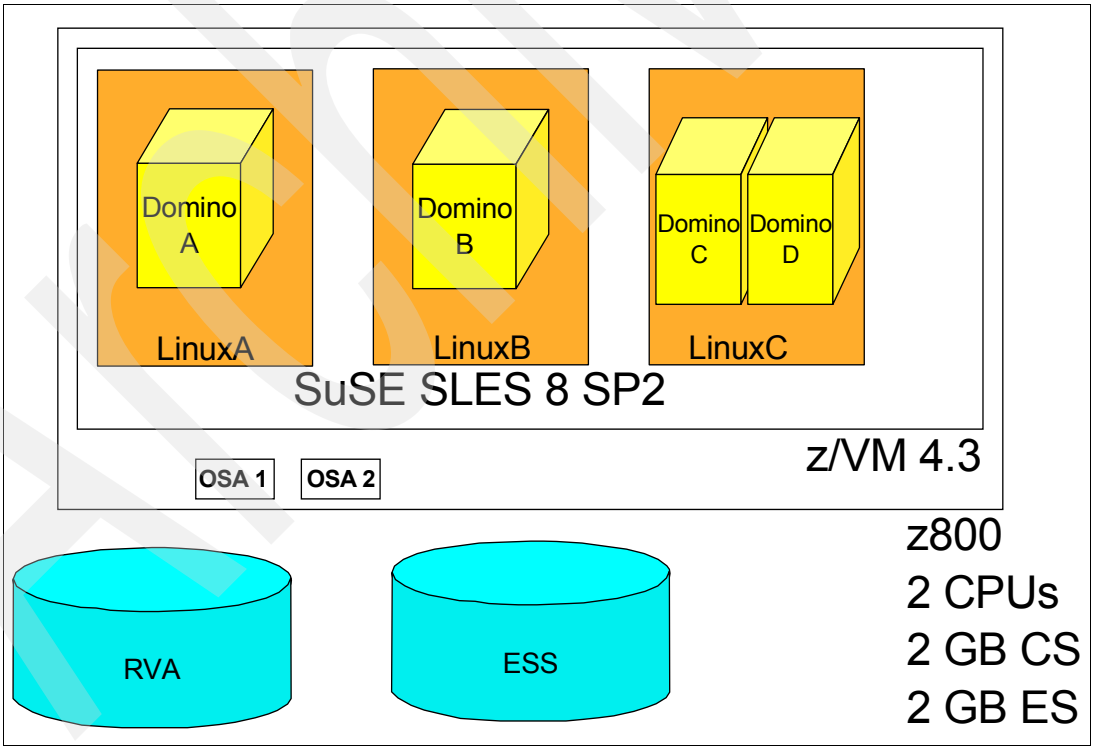



*Figure A-1 Infrastructure for this project*

## Hardware

We used a z800 zSeries with two of the four CPU dedicated to us. It had two gigabytes of central memory and two gigabytes of extended memory. The network connections were established using two OSA cards. The DASD was RVA and ESS.

## Software

For our operating system, we used z/VM 4.3 and SuSE Linux SLES 8 with service pack 2. The Domino server was the Domino for Linux on zSeries 6.5 Beta (M2 and M3, built July 20, 2003 and July 29, 2003). The Notes clients were Notes 6.0, 6.0.2, and 6.5 Beta 2 on various Microsoft Windows 2000 workstations.

Each Linux had one user ID for each Domino server. The systems and user IDs are shown in Table A-1.

*Table A-1 User IDs for Domino*

| | **LinuxA** | **LinuxB** | **LinuxC** |
|---|---|---|---|
| Linux user account | domserva | domservb | domservc, domservd |
| Domino server name | DomServA | DomServB | DomServC, DomServD |

Table A-2 shows a checklist filled in with the values we used in our project.

*Table A-2 The values we used in this project*

| | |
|---|---|
| Domino user account on Linux | domserva, domservb, domservc, domservd |
| Group for Domino user on Linux | notes |
| IP address of your Linux server | 9.10.11.12 |
| Path for Notesdata directory | /domserc#/notesdata |
| Mountpoint for notesdata directory | /domserv# |
| Path for Transactional log | /domserv#/notesdata/translog |
| Mountpoint for transactional log | /domserv#/notesdata/translog |
| Path for mail directory | /domserv#/notesdata/mail |
| Mountpoint for mail directory | /domserv#/notesdata/mail# |
| Domino server name | DomServ# |
| Domino domain name | ITSO |
| DNS entry for Linux and Domino server | Linux#.itso.com |
| Webserver: port ... | 80 |
| DB2 | DB2 @ ? |
| Other applications: DB2, Virus scanner, C/C++ Toolkit, etc. | |

Filesystem for LinuxA:

```
linuxa:~ # df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/dasda1           2.3G  2.0G  243M  89% /
/dev/dasdb1           2.3G  2.0G  334M  86% /opt
/dev/domino/domserva  2.3G  969M  1.2G  45% /domserva
/dev/domino/mail1      16G  3.3G   12G  22% /domserva/notesdata/mail1
/dev/domino/mail2      16G  3.3G   12G  23% /domserva/notesdata/mail2
/dev/domino/transloga
                      2.3G  2.2G  7.9M 100% /domserva/notesdata/translog
shmfs                  93M     0   93M   0% /dev/shm
```

Table A-3 lists the DASD, hardware and filesystem type on each system.

*Table A-3 DASD @ system*

| System | #DASD | Hardware | filesystem type |
|---|---|---|---|
| **LinuxA** | | | |
| / | 1 | | ext2 |
| /opt | 1 | | ext2 |
| /domserva/notesdata/translog | 2 | ESS LVM striped | ext2 |
| /domserva | 1 | ESS LVM striped | ext2 |
| /domserva/mail1 | 7 | ESS LVM striped | ext2 |
| /domserva/mail2 | 7 | ESS LVM striped | ext2 |
| **LinuxB** | | | |
| / | 1 | | ext3 |
| /opt | 1 | | ext2 |
| /domserva/notesdata/translog | 2 | RVA LVM non-striped | ext2 |
| /domserva | 1 | RVA LVM non-striped | ext2 |
| /domserva/mail1 | 7 | RVA LVM non-striped | ext2 |
| /domserva/mail2 | 7 | RVA LVM non-striped | ext2 |
| **LinuxC** | | | |
| / | 1 | | ext2 |
| /opt | 1 | | ext2 |
| /domserva/notesdata/translog | 2 | RVA LVM non-striped | ext2 |
| /domserva | 1 | RVA LVM non-striped | ext2 |
| /domserva/mail1 | 7 | RVA LVM non-striped | ext2 |
| /domserva/mail2 | 7 | ESS LVM non-striped | ext2 |

B

# Linux commands and scripts

In this appendix, we provide the following:

- Useful Linux commands
- A script to add, delete, or show DASD
- A script to add 26 nodes in the /dev/ directory for additional DASD/virtual disk
- A link to the SWAPGEN EXEC, an exec for virtual disk swapping on Linux under VM

# Useful Linux commands

- Run any Linux command by typing the name of the command (and any switches or parameters that are required) at the prompt.
- Unlike DOS and Windows, Linux commands and file names are case sensitive.

Table B-1 provides a list of useful Linux commands.

*Table B-1 Useful Linux commands*

| Command | Syntax | Description |
|---|---|---|
| alias | | Assign name to specified command list |
| cat | *cat file1* | Type or display files |
| cd | *cd /home/user cd ..* | Change directory (same as in DOS) |
| chgrp | *chgrp [option] group fname* | Change group ownership of file or directory |
| chmod | *chmod [option] mode fname* | Change file access permissions |
| chown | *chown [option] owner fname* | Change owner of files or directories |
| cp | *cp file1 /dir1/file2* | Copy file |
| crontab | | Schedules command to run at regularly specified time |
| df | *df -h* | Displays capacity and free capacity on different physical devices.<br>The option -h displays free space in Mb or Gb |
| du | *du directory name* | Displays information about disk usage |
| dir | *dir directory name* | Display directory listing, same as in DOS |
| env | *env* | Display the current environment |
| file | *file <fname>* | Determine file type of fname |
| grep | *grep "pattern to search" fname* | Used to find a string within a file |
| id | *id* | Display user ID and group ID |
| ls | *ls ls -l ls -l *.ini* | Display directory listing, similar to the dir command in DOS |
| man | *man ps* | Displays information from online manual |
| man -k <keyword> | *man -k print* | Lists all manual and commands that relate to <keyword> |
| mv | *mv file1 /dir1/file2* | Move and rename files |
| pwd | *$ pwd* | Print working directory |
| rm | *rm filename* | Remove the file |
| su | *su - username* | Switch user |
| tail | *tail filename* | Display last part of the file |
| tar | *tar options {tar-file-name} {dir-name-to-archive}* | File compression and archiving utility |

| Command | Syntax | Description |
|---|---|---|
| w | *w* | Lists currently logged on users with processing usage |
| which | *which filename* | Displays path to command |
| & | *<command> &* | Starts command as background process |

## The DASD script

This script was originally published in the IBM Redbook *Linux for zSeries and S/390: Large Scale Linux Deployment*, SG24-6824. It allows you to dynamically add, turn on, turn off or list DASD.

```
#!/bin/sh
# dasd - simple utility for dynamic DASD management
if [ "$1" = "add" -a "$2" != "" ]; then
  echo "add range=$2" > /proc/dasd/devices
elif [ "$1" = "on" -a "$2" != "" ]; then
  echo "set device range=$2 on" > /proc/dasd/devices
elif [ "$1" = "off" -a "$2" != "" ]; then
  echo "set device range=$2 off" > /proc/dasd/devices
elif [ "$1" = "list" ]; then
  cat /proc/dasd/devices
else
  echo "Usage: dasd add|on|off vdev_or_range" 1>&2
  echo " dasd list" 1>&2
  exit 2
```

## The mknodDASDaa script

This script adds 26 nodes in the /dev/ directory for additional DASD/virtual disk:

```
#!/bin/sh
# mknodDASDaa - add 26 nodes in the /dev/ directory for additional DASD/Vdisk
let minor=104
for i in a b c d e f g h i j k l m n o p q r s t u v w y z
 do
  for j in ' ' 1 2 3
   do
     mknod /dev/dasda$i$j b 94 $minor
     let minor=minor+1
   done
 done
chmod 660 /dev/dasda*
chgrp disk /dev/dasda*
```

## The COPYDISK EXEC

This REXX EXEC simply copies one quiesced DASD to another:

```
/* copydisk exec */
parse upper arg indisk outdisk .
say "Copy minidisk" indisk "to" outdisk||"?"
pull answer
firstChar = substr(answer, 1, 1)
```

```
if ((answer = 'y') | (answer = 'Y')) then
 do
  queue "SYSPRINT CONS"
  queue "INPUT" indisk "3390"
  queue "OUTPUT" outdisk "3390"
  queue "COPY ALL"
  queue "YES"
  queue "YES"
  queue ""
  DDR
```

## The SWAPGEN EXEC

The SWAPGEN EXEC generates a virtual disk swap for Linux on VM. The SWAPGEN EXEC is available on the Internet:

http://sinenomine.net/node/265

C

# Educational material

In this appendix, we list courses and material you may be interested in which will help you build your skills in various areas of Domino 6.5 for Linux on zSeries.

# IBM Education courses

Available on the Internet at:

http://www-3.ibm.com/services/learning/index.html

## z/VM and Linux for zSeries

- Installing, Configuring and Servicing z/VM for Linux Guests, ZV06G
- z/VM & Linux Connectivity and Management, ZV100
- Linux for zSeries running under z/VM: Advanced Topics Workshop, LXOZM
- Advanced Solutions for Linux on zSeries, ZL150
- Installing, Configuring and Servicing z/VM for Linux Guests, ZV060
- Linux Basics - A zSeries and S/390 Perspective, HLX13
- Linux Implementation for zSeries and S/390, ZL100
- Linux on zSeries for Solaris Administrators, QLX18
- z/VM and Linux Connectivity and Management, ZV100
- Satellite Linux Basics: An S/390 Perspective (ILS-DVL), H13S0

## Linux

- QLX01 - Linux Introduction: What is it? Who's using it? And why?
- QLX02 - Linux Basics and Installation
- QLXA2 - Linux Basics and Installation (self-paced virtual class)
- QLX03 - Linux System Administration
- QLXA3 - Linux System Administration (self-paced virtual class)

## Domino

- LA760G - Administering IBM Lotus Domino 6: Building the Infrastructure
- LA770G - Administering IBM Lotus Domino 6: Managing Servers and Users
- LA750G - Administering IBM Lotus Domino 6: Operating Fundamentals
- LD510G - Developing IBM Lotus Domino 6 Applications: Foundation Skills
- LD520G - Developing IBM Lotus Domino 6 Applications: Intermediate Skills
- LD720G - Exploring New Features in IBM Lotus Domino 6 Administration
- LD420G - Exploring the New Features of IBM Lotus Domino Designer 6
- ES72G - Implementing Domino
- LX01G - Getting Started with Linux and IBM e-business Software™ on IBM e-servers

D

# Install script.dat

In this appendix, we show a sample script for installing three partitioned Domino servers. We have highlighted all the lines you should review and alter for your installation.

Refer to Chapter 7, “Domino installation” on page 133 for more information about the different parameters. You will also find a description of the script mode installation in the Install guide for Domino 6.5 for Linux on zSeries.

```
#
# Lotus Domino Server for UNIX
# Install Script Template
#
# History of the modification of the script_rev
# 05/27/1999;script_rev=2; Added for allowing unique ownership
#           for partitioned server data dirs;
# 05/01/2000;script_rev=3; Added ddirs option for unix installer;
# 09/26/2001;script_rev=4; Added optional iNotes Web Access of unix
#           installation and Incremental Installer
#           from V5.09 for OS AIX and Solaris;
# 09/27/2001;script_rev=5; It is used for Domino 6 unix installation.
# 11/07/2001;    Added two installation control parameters.
#           opt_lotus_softlink is used for adding soft linking from program
#           directory to /opt/lotus;
#           template_install_option is used for installing template files
#           by option;
# 01/11/2002;script_rev="5.1"; For identifying the new features of opt_lotus_softlink and
#           template_install_option variables in Domino 6.
#
# 03/01/2002;script_rev="5.2";   Added asp_install_option control parameter,
#           which is used for optional setting up an ASP server refers to the
#           configuration of an Application Service Provider server.
#
#04/20/2002; script_rev="5.3"; Added add_data_directories_only control parameter,
#           which is used for adding additional data directories to your existing
#           Domino installation.
#
#06/19/2002; script_rev="5.4"  Added the server descriptions for the different
#           installation type.
#
```

```
#08/07/2002; script_rev="5.5"   Modified server names.
#
#03/17/2003; script_rev="5.6"   added server setup or restart server feature
#            after single local server is installed
script_rev="5.6"  # please do not change from user. otherwise it will break your
installation
#
# There are two ways to use the Lotus Domino for UNIX Install Program:
#     1) Interactive Mode - you are prompted for the installation settings
#     2) Script Mode - you specify the installation settings in a script file
#
# Interactive Mode
#     To use, invoke the install program without any arguments,
#     such as "./install".  The Install Program will prompt you for all
#     settings necessary to install the Domino Server on the same
#     computer that you are running "install" on.  To use Interactive mode,
#     you do not need this template file.
#
# Script Mode
#     To use, copy this file, edit it, and invoke the install program
#     such as "./install -script /tmp/script.dat".  To edit the script
#     file, read each section that follows, and change the configuration
#     setting as necessary, following the syntax specified for that section.
#     Script mode allows you to install the Domino Server
#     to one or more computers, each with a different
#     installation configuration as necessary.
#
#
#
#
# "target_hosts"
#
#    If not set (by default it is commented-out),
#    the Domino Server will be installed only to the computer on which
#    the Install Program is being run (the "local host").  The installation
#    settings used will be those of the "Default Settings" sections following
#    this section.
#
#    If set, specifies one or more target hosts on which the Domino Server
#    will be installed.  The local host will be installed to only if it is
#    one of the hosts specified in the target_hosts list.  For each host, the
#    installation settings used will be those of the "Default Settings"
#    sections, unless there is a "custom_host_settings" entry for a particular
#    host.  The "custom_host_settings" section follows the "Default Settings"
#    sections in this file, and allows you to override one or more of the
#    default settings for any host in the target_hosts list.
#
#    The double-quoted, comma-separated list may be specified on
#    a single line, or using multiple lines.
#    For example:
#    target_hosts = (
#      "host1",
#      "host2",
#      "host3"
#    )
#
#    To use this setting, be sure to remove the '#' comment character
#    on the "target_hosts=" line.
#
# !!!
```

```
#target_hosts=("host1","host2")
#####################################################################
#
# begin "Default Settings" sections
#
#
#
#
# "installation_type"
#
#There are different types of Domino server setups available.
#Your choice depends on your needs, and licensing options.
#
#The installation types are:
#    * Domino Messaging Server:
#  Select this option to install a Domino Server
#  that provides messaging services.
#  Note that it does not include support
#  for application services or Domino clusters.
#
#    * Domino Utility Server:
#  Select this option to install a Domino Server
#  that provides application services
#  only, with support for Domino clusters. Note that it does
#  not include support for messaging services.
#  The Domino Utility Server is a new installation type for
#  Lotus Domino 6 that removes client access license requirements.
#  See full licensing text for details from UI installation.
#
#    * Domino Enterprise Server:
#  Select this option to install a Domino Server
#  that provides both messaging and
#  application services, with support for Domino clusters.
#
# Specifies the set of product options that are to be installed.
#
#
# Specify the desired "installation_type" by its number.
#
# Domino Messaging Server   : 0
# Domino Utility Server    : 1
# Domino Enterprise Server  : 2
#
# !!!
installation_type = 2
#
#
# "template_install_option"
#
#     The optional installation feature for template files is designed for
#     users who are installing over a previous version of the Domino Server
#     and wish to keep all the previous template files. If this is not an
#     installation over an existing Domino Server, all template files must
#     be installed.
#    This control variable "template_install_option" is set to 1 or 0.
#    template_install_option = 1;which means installing template files;
#               The default value is set to 1
#    template_install_option = 0;which means not installing template files;
template_install_option = 1
#
```

```
#
# "asp_install_option"
#
#  The option to setup an ASP server refers to the configuration of an
#  Application Service Provider server. This type of server can only be
#  configured after an Enterprise Server installation.
#  Setting asp_install_option = 1 below will cause the Domino Setup program
#  to configure the server appropriately for ASP functionality.
#  This will add security features not present in a normal configuration,
#  so do not set asp_install_option = 1 unless an ASP configuration is
#  specifically required for this server.
#  The dafault value will be set to 0 for all servers installation.
#
asp_install_option = 0
#
#  "start_server_setup"
#    The start_server_setup option is for single local domino server only.
#    Do not use this option for partitioned server or remote server installation.
#    For new server installations, this option allows server setup to be launched
#    automatically after the installation is complete.
#
#    For server upgrades, this option allows the server to restart automatically
#    after the installation is complete.
#
#       start_server_setup = 0;<---> manual server setup.
#       start_server_setup = 1;<---> launches local server setup or restarts server
#       start_server_setup = 2;<---> launches server setup in listen mode
#                 for remote server setup
#
#    The default value of start_server_setup is 0, which does not launch
#    server setup or restart the server after installation.
#
#    Change the value of start_server_setup to 1 to launch server setup
#    after a new server installation or to restart the server after a server upgrade.
#
#    Change the value of start_server_setup to 2 to launch server setup
#    in listen mode for new server installations.  You will then be able to connect
#    to the server with the Remote Server Setup tool.
#
start_server_setup = 0
#
#
# "add_data_directories_only"
#
# Set this setting to 1 if you only wish to add additional data directories
# to your existing Domino installation.  You do not need to specify
# existing data directories when using this option.  Note, however, that
# both the "program_directory" &
# "installation_type" settings must be set correctly so that the proper data
# is installed.  Specify the new data directories to be installed
# as described in the "data_directories" section, below.
#
```

Refer to 7.8, “Adding a partitioned Domino server (DPAR)” on page 167 for more information about how to add additional Domino servers to your installation. Change the value for the parameter below to 1 if you have already installed at least one Domino server on your system and now want to add only one or more partitioned servers.

```
add_data_directories_only = 0
#
#
```

```
# "program_directory"
#
# Specifies the path to which the Domino program directories & files
# are to be installed.  The top directory of the program tree is always
# "lotus", and will be appended to the specified path if not supplied.
# If the specified path is not /opt/lotus, a link will be created from
# /opt/lotus to the program tree.
# This path must be double-quoted.
#
# !!!
program_directory = "/opt/lotus"
#
#
#    The Domino 6 software no longer requires the /opt/lotus soft link for
#    handling multi- Domino Server installations on a single operating system.
#    You can now select your own option for the /opt/lotus soft link feature
#    if you have chosen a directory other than /opt/lotus for installation.
#   opt_lotus_softlink = 0; which means not creating "/opt/lotus" soft link;
#                       its default  is set to 0;
#   opt_lotus_softlink = 1; which means creating "/opt/lotus" soft link;
opt_lotus_softlink = 0
#
#
# "data_UNIX_user"
#
# Specifies the (default) Domino UNIX user which will own the
# Domino Data Directories, and will be used to run the Domino Server.
# The "data_UNIX_user" setting may be overridden for a particular
# data directory in the "data_directories" or "custom host settings"
# sections, below.
# Please note that the Domino program files will be owned by the system,
# and that this setting affects only the Domino Data Directories
#
# !!!
# data_UNIX_user = "notes"
#
#
# "data_UNIX_group"
#
# Specifies the (default) Domino UNIX group which will own the
# Domino Data Directories.
# The Domino UNIX user must be a member of this group.
# The "data_UNIX_group" setting may be overridden for a particular
# data directory in the "data_directories" or "custom host settings"
# sections, below.
# Please note that the Domino program files will be owned by the system,
# and that this setting affects only the Domino Data Directories
#
# !!!
# data_UNIX_group = "notes"
#
#
# "data_directories"
#
# Defines settings for one or more Domino Data Directories to be installed.
#
# You would define more than one data directory here if you want to run
# multiple Domino Partitioned Servers on the same computer.
#
# You must define at least one data directory.  In the "data_directories:"
```

```
# entry below, edit the path "/local/notesdata" as necessary.
# To define more than one data directory, copy and paste the
# "data_directories:" entry below (from the line begining with
# "data_directories:", to the line containing the '}' character) as many
# times as necessary, one for each data directory.  Then edit the
# path (must be double-quoted) and/or owner settings for each.
```

If you want to install more than one Domino server using this script, like described in 7.8, “Adding a partitioned Domino server (DPAR)” on page 167 you must have as many data_directory definitions as you want to install Domino servers. To install domserva, domservb and domservc this part of the script look like:

```
#
# Begin server directory definitions
#
# Definitions for domserva:
data_directories: "/domserva/notesdata" {
    # un-comment either of the following lines to override the defaults
data_UNIX_user  = "domserva"
data_UNIX_group = "notes"
}
#
# Definitions for domservb:
data_directories: "/domservb/notesdata" {
    # un-comment either of the following lines to override the defaults
data_UNIX_user  = "domservb"
data_UNIX_group = "notes"
}
#
# Definitions for domservc:
data_directories: "/domserva/notesdatc" {
    # un-comment either of the following lines to override the defaults
data_UNIX_user  = "domservc"
data_UNIX_group = "notes"
}
#
# End server directory definitions
#
#
#
# end "Default Settings" sections
#
#####################################################################
#
#
#
# "custom_host_settings"
#
#    When the "target_hosts" setting is used to install to multiple hosts,
#    the settings from the "Default Settings" sections above are used for
#    for each host, unless there is a "custom_host_settings" entry
#    specified for a particular host.  The "custom_host_settings" entry allows
#    you to override one or more of the install settings for a particular host.
#    You may have zero or more "custom_host_settings" entries; one for each
#    host for which you need to override the default settings.
#
#    To make a "custom_host_settings" entry, copy the example entry below
#    from the first line ("#custom_host_settings: TargetHostname {") to the
#    last ("}").  Change "TargetHostName" in the first line to the hostname
#    for which you need to override default settings.  Then remove the
```

```
#    comment character ("#") from the first and last lines, and from
#    setting lines you need to override.  Now edit those settings as
#    described in the "Default Settings" sections above.  Repeat for
#    as many hosts as necessary.
#
# !!!
#custom_host_settings: TargetHostname {
#    installation_type  = 0
#    template_install_option=1
#    opt_lotus_softlink=0
#    program_directory  = "/opt/lotus"
#    data_UNIX_user     = "notes"
#    data_UNIX_group    = "notes"
#    # add additional "data_directories:" entries as necessary, as
#    # explained in the "data_directories:" section above
#    data_directories: "/local/notesdata" {
#        # un-comment either of the following lines to override the defaults
#        # data_UNIX_user  = "notes"
#        # data_UNIX_group = "notes"
#    }
#}
```

E

# Creating virtual disks manually

In this appendix, we describe how to create virtual disks manually. In 6.6, “Set up swap space” on page 125, we describe how to use the SWAPGEN EXEC; that exec can be found at:

http://sinenomine.net/node/265

There are two ways to create a virtual disk: in the user’s directory entry, or by users themselves.

## User’s directory entry

The following example is a sample of a user directory entry for virtual disk definition. This directory entry will create 10 virtual disks, each with 50000 blocks reserved for them:

```
MDISK 0301 FB-512 V-DISK 50000 MR
MDISK 0302 FB-512 V-DISK 50000 MR
MDISK 0303 FB-512 V-DISK 50000 MR
MDISK 0304 FB-512 V-DISK 50000 MR
MDISK 0305 FB-512 V-DISK 50000 MR
MDISK 0306 FB-512 V-DISK 50000 MR
MDISK 0307 FB-512 V-DISK 50000 MR
MDISK 0308 FB-512 V-DISK 50000 MR
MDISK 0309 FB-512 V-DISK 50000 MR
MDISK 0310 FB-512 V-DISK 50000 MR
```

## Created by user

The following example shows how a user can create a virtual disk when given sufficient resources to do so.

```
DEFINE VFB-512 AS 323 BLK 50000
DASD 0323 DEFINED
```

## Format the virtual disk

Use the CMS format command to format the disk to a block size of 512.

```
format 323 b (blksize 512
DMSFOR603R FORMAT will erase all files on disk B(323). Do you wish to continue?
Enter 1 (YES) or 0 (NO).
1
DMSFOR605R Enter disk label:
LNXSWP
Formatting disk B
50000 FB-512 blocks formatted on B(323)
q disk b
LABEL  VDEV M  STAT   CYL TYPE BLKSZ   FILES  BLKS USED-(%) BLKS LEFT  BLK TOTAL
LNXSWP 323 B   R/W    FB 9336  512        0         32-00      49968      50000
```

**Note:** Ensure that the correct block size is used on the format command. The diagnose driver will only handle a virtual disk with a block size of 512. The default is 1024.

## Reserve swap space on virtual disk

Use the CMS reserve command to preallocate the swap space on the virtual disk, as follows:

```
RESERVE LINUX SWAP b
DMSRSV603R RESERVE will erase all files on disk B(323). Do you wish to continue
 Enter 1 (YES) or 0 (NO).
1
Reserving disk B
Ready; T=0.01/0.01 09:50:32
q disk b
LABEL  VDEV M  STAT   CYL TYPE BLKSZ   FILES  BLKS USED-(%) BLKS LEFT  BLK TOTA
LNXSWP 323 B   R/W    FB 9336  512        1      50000-99          0      5000
Ready; T=0.01/0.01 09:50:41
listfile * * b (date
FILENAME FILETYPE FM FORMAT LRECL       RECS     BLOCKS   DATE     TIME
LINUX    SWAP     B6 F        512      49574      49574  8/19/03  9:50:32
```

Now the virtual disk is 99% full with a single file on it called LINUX SWAP. Notice that the swap file LRECL is the same size at the BLKSZ of the disk, and that they are both 512.

# Abbreviations and acronyms

| | |
|---|---|
| **ACL** | access control list |
| **ASP** | application service provider |
| **CHPID** | channel path identifier |
| **CMS** | Conversational Monitor System |
| **CP** | control program |
| **DASD** | direct access storage device |
| **DCR** | Data Connection Resource |
| **DECS** | Domino Enterprise Connection Services |
| **DNS** | Domain Name Services |
| **DPA** | dynamic paging area |
| **DPAR** | Domino partitioned server |
| **ESS** | Enterprise Storage Server |
| **FTP** | File Transfer Protocol |
| **IBM** | International Business Machines Corporation |
| **IFL** | Integrated Facility for Linux |
| **IML** | Initial Machine Load or Initial Microcode Load |
| **ITSO** | International Technical Support Organization |
| **IUCV** | Inter-User Communications Vehicle |
| **LAN** | local area network |
| **LDAP** | Lightweight Directory Access Protocol |
| **LPAR** | logical partition |
| **LVM** | logical volume manager |
| **MDC** | minidisk cache |
| **MHz** | megahertz |
| **MIP** | millions of instructions per second |
| **MQLSX** | MQSeries Lotus Script Extension |
| **NRPC** | Notes Remote Procedure Call |
| **OSA** | Open Systems Adapter |
| **PV** | physical volume |
| **RAID** | redundant array of independent disks |
| **RAMAC®** | RAID Architecture with Multi-Level Adaptive Cache |
| **RVA** | RAMAC virtual array |
| **SCSI** | small computer system interface |
| **SNMP** | simple network management protocol |
| **VG** | volume group |

# Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

## IBM Redbooks

For information on ordering these publications, see “How to get IBM Redbooks” on page 399. Note that some of the documents referenced here may be available in softcopy only.

### Linux

- *Linux on IBM @server zSeries and S/390: Performance Measurement and Tuning*, SG24-6926
- *Linux on IBM @server zSeries and S/390 : ISP/ASP Solutions*, SG24-6299
- *Linux on IBM @server zSeries and S/390 : Large Scale Linux Deployment*, SG24-6824
- *Linux for IBM @server zSeries and S/390: Distributions*, SG24-6264
- *Linux on IBM @server zSeries and S/390: System Management,* SG24-6820
- *Linux for S/390*, SG24-4987

### VM

- *Linux on IBM @server zSeries and S/390: TCP/IP Broadcast on z/VM Guest LAN,* REDP-3596
- *Linux on IBM @server zSeries and S/390: High Availability for z/VM and Linux*, REDP-0220
- *Linux on IBM @server zSeries and S/390: Building SuSE SLES8 Systems under z/VM,* REDP-3687
- *Partitioning DASD for Linux Guests Running under z/VM,* TIPS0277

### Domino

- *Lotus Domino 6 for Linux*, SG24-6835
- *Lotus Domino 6 Spam Survival Guide for IBM @server,* SG24-6930
- *Upgrading to Lotus Notes and Domino 6*, SG24-6889
- *Domino Designer 6: A Developer’s Handbook*, SG24-6854 (for information about DCRs)
- *TurboLinux Integration Guide for IBM @server xSeries and Netfinity,* SG24-5862
- *Domino and WebSphere Together,* Second Edition, SG24-5955
- *Lotus Domino for S/390: Running a Large Domino System,* SG24-5984

### Tivoli

- *IBM Tivoli Storage Management Concepts,* SG24-4877

# Other publications

These publications are also relevant as further information sources:

- *z/VM CP Planning and Administration*, SC24-6043
- *z/VM CP Command and Utility Reference*, SC24-6008
- *z/VM TCP/IP Planning and Customization*, SC24-6019
- *IBM Tivoli Storage Manager for Mail: Data Protection for Lotus Domino for UNIX and OS/400 Installation and User's Guide*, SC32-9056
- *z/VM Guide for Automated Installation and Service, Version 4, Release 4.0*, GC24-6064

# Online resources

These Web sites and URLs are also relevant as further information sources:

- Linuxvm.org - The Linux on zSeries portal

  http://linuxvm.org
- DeveloperWorks - IBM Boeblingen

  http://www10.software.ibm.com/developerworks/opensource/linux390/index.shtml
- ISV applications for Linux on zSeries

  http://www.ibm.com/servers/eserver/zseries/solutions/s390da/linuxproduct.html
- The IBM Web site on z/VM and Linux

  http://www.vm.ibm.com/linux
- z/VM publications

  http://www.vm.ibm.com/pubs/
- Lotus developers domain, including product documentation

  http://www.lotus.com/ldd
- IBM Lotus home page

  http://www.lotus.com
- SuSE, a developer of Linux

  http://www.suse.de
- UnitedLinux, a private company equally owned by four Linux development companies

  http://www.unitedlinux.com

## Newsgroups

The linux-390 list server is where the community meets. To subscribe to it, send an e-mail to listserv@vm.marist.edu with this line in the body:

```
subscribe linux-390
```

You should receive a return e-mail in 5 - 15 minutes, asking you to click a URL. When you click that URL, you're in. You'll get another, confirmation e-mail. Save this confirmation e-mail, as it contains many instructions. Then you can send the same address an e-mail to tailor your subscription.

Here are the most useful commands:

```
set linux-390 nomail    // stay subscribed get no e-mail (good for holidays)
set linux-390 mail      // turn e-mail back on
signoff linux-390       // when you've had enough
set linux-390 digest    // get one big e-mail a day
set linux-390 repro     // get a copy of your own appends
get linux-390 log0204   // get a month worth of appends (e.g. April 2002)
```

To append to the list, simply send an e-mail to linux-390@vm.marist.edu. The linux-390 archives are on the Internet at:

http://www.marist.edu/htbin/wlvindex?linux-390

There is also a SuSE Linux mailing list archive regarding Domino:

http://lists.suse.com/archive/suse-domino/

# How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

**ibm.com**/redbooks

# Help from IBM

IBM Support and downloads

**ibm.com**/support

IBM Global Services

**ibm.com**/services

# Index

## Symbols



## Numerics



## A



## B



## C





## D

## E



## F



## G



## H



## I

## J



## K



## L



## M

## N



## O



## P

## Q



## R



## S

## T



## U



## V

## W



## X



## Y



## Z

# IBM Lotus Domino 6.5 for Linux on zSeries Implementation

(0.5” spine)
0.475”<->0.873”
250 <-> 459 pages

# IBM Lotus Domino 6.5 for Linux on zSeries Implementation

**Planning the environment - operating system, disk, network**

**Installing and administering Linux and Domino**

**Capacity planning and performance**

This IBM® Redbook will help the reader plan for, install, and configure the new IBM Lotus® Domino® 6.5 product on an IBM @server® zSeries® system running Linux. We discuss why you should consider running Domino for Linux on zSeries, and list the advantages of running Linux in a guest under z/VM®.

We then describe the structure of running Domino for Linux on zSeries for those who are not familiar with all of those products, and review what is new in Domino 6.5, since this is the first release that is supported on Linux on zSeries.

We provide detailed technical information about planning, allocating, and managing disk space, network considerations, installing Linux and Domino, and administering Domino. We also discuss systems management, capacity planning and performance tuning, connectivity to DB2®, migration from previous Domino releases or Domino servers on other platforms, and troubleshooting.

This redbook is targeted to zSeries systems programmers, Domino administrators, consultants, and service technicians.

This updated version includes information from customer and IBM experiences.

**INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION**

**BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE**

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

**For more information:**
**ibm.com**/redbooks

SG24-7021-01 ISBN 0738491748