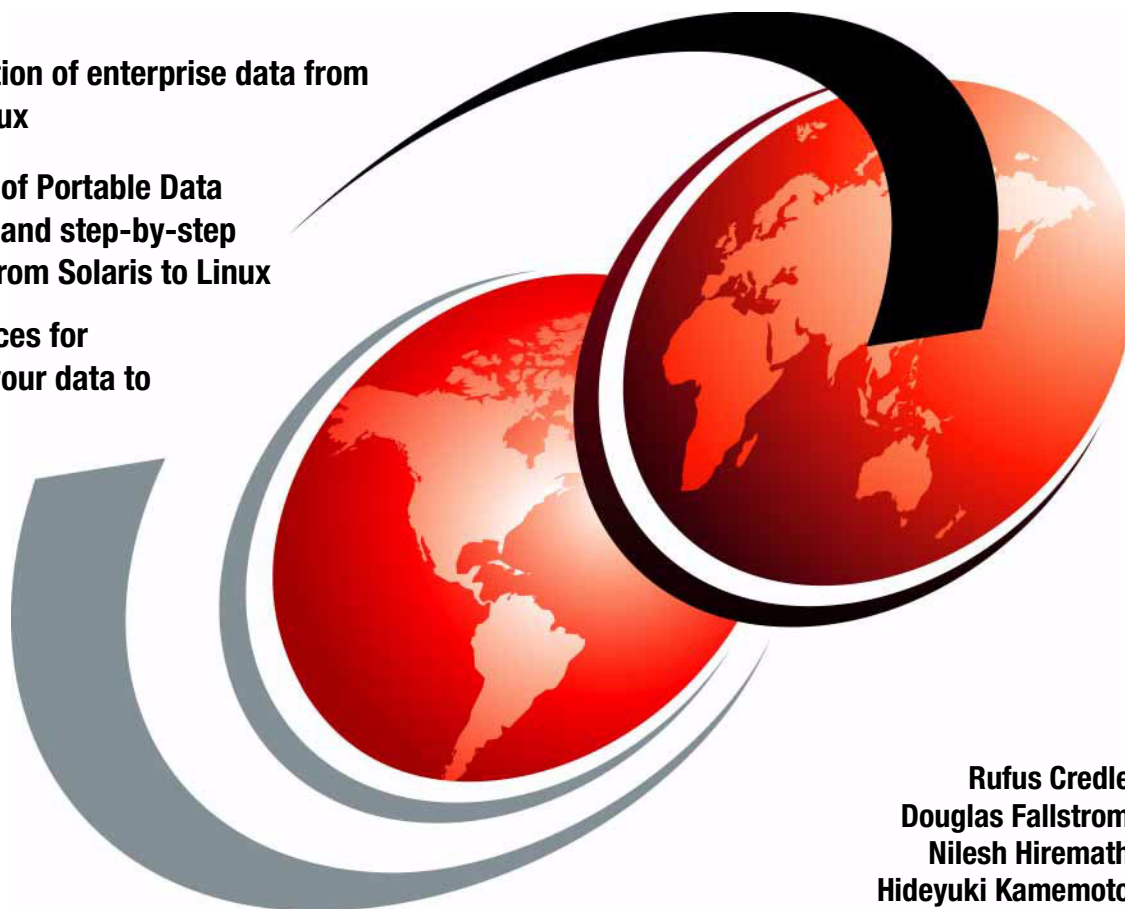


# Migrating to Linux Using VERITAS Portable Data Containers on IBM @server xSeries and BladeCenter

Fast migration of enterprise data from  
UNIX to Linux

Mechanics of Portable Data  
Containers and step-by-step  
migration from Solaris to Linux

Best practices for  
migrating your data to  
Linux



Rufus Credle  
Douglas Fallstrom  
Nilesh Hiremath  
Hideyuki Kamemoto





International Technical Support Organization

**Migrating to Linux Using VERITAS Portable Data  
Containers on IBM @server xSeries and  
BladeCenter**

December 2004

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

**First Edition (December 2004)**

This edition applies to VERITAS Storage Foundation 4.0, Oracle 9i and DB2 8.1.

**© Copyright International Business Machines Corporation 2004. All rights reserved.**

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

# Contents

<b>Notices</b> .....	vii
Trademarks .....	viii
<b>Preface</b> .....	ix
The team that wrote this redbook .....	ix
Become a published author .....	xi
Comments welcome .....	xi
<b>Chapter 1. The growing acceptance of Linux</b> .....	1
1.1 Why migrate? .....	2
1.2 IBM @server BladeCenter and xSeries commitment to Linux .....	3
1.3 VERITAS commitment to Linux .....	4
1.4 VERITAS solutions .....	4
1.4.1 VERITAS Storage Foundation for Linux .....	4
1.4.2 VERITAS NetBackup for Linux .....	5
1.4.3 VERITAS Volume Replicator for Linux .....	5
1.4.4 VERITAS Cluster Server for Linux .....	6
1.4.5 VERITAS Opforce for Linux .....	6
1.4.6 VERITAS i3 for Linux .....	6
1.5 A world-class community of enterprise Linux partners .....	6
<b>Chapter 2. UNIX to Linux migration considerations</b> .....	9
2.1 Complexity of migrations .....	10
2.2 Migration challenges .....	10
2.2.1 Operating system dependence .....	10
2.2.2 File management dependence .....	13
2.2.3 Endianess .....	14
2.2.4 Different techniques of migrating data .....	15
<b>Chapter 3. VERITAS Storage Foundation</b> .....	17
3.1 VERITAS Storage Foundation overview .....	18
3.2 Quality of Storage Services .....	19
3.3 Multi Volume File System .....	21
3.4 Dynamic Multi Pathing .....	21
3.5 Intelligent Storage Provisioning .....	22
3.6 FlashSnap .....	23
3.7 Portable Data Containers .....	24
<b>Chapter 4. VERITAS Portable Data Containers for migration</b> .....	25

4.1	The technical barriers that Portable Data Containers overcome . . . . .	26
4.1.1	Operating system . . . . .	26
4.1.2	Volume Manager. . . . .	27
4.1.3	File system . . . . .	28
4.1.4	Applications. . . . .	29
4.1.5	Problems that Portable Data Containers do not address . . . . .	29
4.2	Step-by-step migration example . . . . .	30
4.2.1	Lab environment . . . . .	30
4.2.2	VERITAS Storage Foundation 4.0 installation and configuration . . .	33
4.2.3	VERITAS Storage Foundation 4.0 post installation . . . . .	42
4.2.4	VERITAS Enterprise Administrator . . . . .	44
4.2.5	Initializing disks for a new file system . . . . .	49
4.2.6	Starting with an existing Volume Manager or File System setup . . .	60
4.2.7	Step-by-step migration . . . . .	61
	<b>Chapter 5. Performance</b> . . . . .	67
5.1	Data migration performance . . . . .	68
5.1.1	Data collection. . . . .	69
5.2	Scalability . . . . .	69
5.2.1	Scalability performance results . . . . .	70
5.2.2	CPU and I/O . . . . .	71
5.3	Throughput . . . . .	73
5.3.1	Throughput results . . . . .	74
5.3.2	CPU and I/O . . . . .	75
5.4	Performance comparison . . . . .	76
5.5	Other performance considerations . . . . .	78
5.6	Conclusion. . . . .	78
	<b>Chapter 6. Best practices</b> . . . . .	79
6.1	Minimizing data migration surprises . . . . .	80
6.1.1	Preventing the file system from writing incompatible data . . . . .	80
6.2	Making the conversion process safer and faster. . . . .	81
6.3	Calculating the space requirements of the conversion process . . . . .	81
6.4	Unconverting a file system . . . . .	82
6.5	Using VERITAS FlashSnap for a test migration . . . . .	83
6.5.1	Requirements and recommendations . . . . .	83
6.5.2	Migration example environment . . . . .	85
6.5.3	Situation before migration . . . . .	86
6.5.4	Summary of migration steps . . . . .	88
6.5.5	Migration example steps . . . . .	92
6.6	Oracle application migration scenario . . . . .	100
6.7	DB2 application migration . . . . .	118
	<b>Appendix A. Manual Pages</b> . . . . .	123

VERITAS File System, fsadm_vxfs .....	123
Maintenance command: fscdsconv(1M).....	135
<b>Abbreviations and acronyms</b> .....	<b>137</b>
<b>Related publications</b> .....	<b>139</b>
IBM Redbooks .....	139
Other publications .....	139
Online resources .....	139
How to get IBM Redbooks .....	140
Help from IBM .....	140
<b>Index</b> .....	<b>141</b>





# 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®

@server®

Redbooks (logo) ™

ibm.com®

xSeries®

AIX®

BladeCenter™

Domino®

DB2®

Everyplace®

Informix®

IBM®

POWER™

Redbooks™

Tivoli®

TotalStorage®

WebSphere®

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

Rational®

The following terms are trademarks of other companies:

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

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

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

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

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

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

# Preface

This IBM® Redbook discusses Portable Data Containers. Portable Data Containers are a tool that allows customers to migrate their applications from UNIX® to Linux® more seamlessly. They are a part of VERITAS Storage Foundations 4.0. With Portable Data Containers, a customer can migrate an application without moving that application's data.

This redbook describes the concepts of storage management and the key features of VERITAS Storage Foundations 4.0 and . In addition, it discusses a number of considerations that you should review prior to a UNIX to Linux migration, and then takes you through a step-by-step migration from Solaris to Linux. It discusses the performance and scalability of Portable Data Containers and compares the throughput of Portable Data Containers with theoretical network throughput and disk transfer speeds. Finally, this redbook includes best practices for migrating applications to Linux using Portable Data Containers. In particular, it identifies the steps that are required to migrate an Oracle or DB2® database from a UNIX server to a Linux server. With this knowledge, customers can accelerate their migration plans to Linux to take advantage of the cost benefits provided by the open-source operating system.

The audience for this redbook are IBM Clients and Business Partners as well as IT specialists and database administrators.

## The team that wrote this redbook

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



**Rufus Credle** is a Certified Consulting I/T Specialist at the ITSO, Raleigh Center. In his role as project leader, he conducts residencies and develops Redbooks™ about network operating systems, ERP solutions, voice technology, high availability and clustering solutions, Web application servers, pervasive computing, and IBM and OEM e-business applications, all running IBM @server™ xSeries® and IBM @server BladeCenter™ systems. Rufus's various positions during his IBM career have included assignments in administration and asset management, systems engineering, sales and marketing, and IT services. He holds a BS degree in business management from Saint Augustine's College. Rufus has been employed at IBM for 24 years.



**Douglas Fallstrom** is a Senior Linux Technical Product Manager at VERITAS Software where he manages the technical strategy for the Linux platform. He is responsible for the technical requirement definitions for all Storage Foundation products on Linux. Douglas also works with performance benchmarks on Linux as well as VERITAS Cluster File System and Storage Foundation for Oracle RAC. Douglas has been with VERITAS for nearly three years. Prior to joining VERITAS, Douglas spent seven years at Sun Microsystems.



**Nilesh Hiremath** is a Software Engineer with VERITAS Software in Pune, India. He has 10 years of experience in the storage and system administration field. He holds a degree in M.Tech from IIT Kanpur India and has worked at VERITAS Software for four years. His areas of expertise include system administration, performance analysis, and electric power systems.



**Hideyuki Kamemoto** is an IT Specialist at the Linux Support Center of IBM Japan. He has four years of experience in the Linux field and ten years of experience in UNIX fields, such as Solaris and HP-UX. He has provided technical support for IBM @server xSeries. His areas of expertise include Solaris, HP-UX, X Window System, VERITAS Storage Foundation, VERITAS Cluster Server, Linux, and IBM @server xSeries.

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

Cecilia Bardy, Tamikia Barrow, and Diane O'Shea  
ITSO, Raleigh Center

Amy Freeman, Brand and ISV Communications, xSeries Linux Marketing  
Jeff Huxtable, SSG Alliance Relationships  
IBM Research Triangle Park

Ram Pandiri, Senior Engineering Program Manager  
VERITAS Austin

Ahmad Filsoof, Business Development Manager  
HanCheng Hsiung, Sr. Development Manager, Storage Foundation for DB2  
Raghu Malige, Principal Engineer  
Lalgudi Sujatha, Sr. Software Engineer, Storage Foundation for DB2  
VERITAS Software, Mountain View, CA

Susan Encinas, ISG Marketing Support  
IBM Tucson

## 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](http://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](http://ibm.com/redbooks)

- ▶ Send your comments in an email to:

[redbook@us.ibm.com](mailto:redbook@us.ibm.com)

- ▶ Mail your comments to:

IBM Corporation, International Technical Support Organization  
Dept. HQ7 Building 662  
P.O. Box 12195  
Research Triangle Park, NC 27709-2195





# The growing acceptance of Linux

The Linux operating system is being used in production environments for a wide variety of applications. This chapter discusses the growing acceptance of Linux and why customers are migrating to Linux from other operating systems. It also highlights the commitment of IBM and VERITAS to Linux.

## 1.1 Why migrate?

Over the past few years, the Linux operating system has become a real and viable alternative for UNIX users by delivering the power and flexibility of a UNIX server or desktop at competitive prices. Some of the features of the Linux operating system are:

- ▶ A fully functional operating system similar to a UNIX system. Linux has all the standard features of enterprise UNIX systems. The four fundamental components of Linux are:
  - Kernel: Provides program and hardware device management (such as disks and printers).
  - Shell: Receives commands from the user and sends those commands to the kernel for the execution through a standard user interface.
  - File structure: Files are organized into directories with the disk hardware. Each directory may contain any number of subdirectories each holding files. This is a similar structure to PC operating system file structures.
  - Utilities: Tools to help manage the environment and the applications running in the server.
- ▶ The same multiuser and multitasking capabilities as large UNIX servers.
- ▶ The same level of system administration that you find on standard UNIX systems. Users can run several programs concurrently. You can create user accounts for different users and define their access rights to the files and system components. The Linux operating system also provides installation of new devices, network connection, and control.
- ▶ A powerful set of development tools for creating industrial-strength applications.

Overall, there is a wide range of support for Linux across the entire spectrum of IT system requirements. Table 1-1 summarizes some of the key business requirements and the applications that address them.

*Table 1-1 Business areas and solutions*

<b>Infrastructure solutions</b>	<b>Vendor or solution</b>
Database and Application Development Tools	DB2 UDB, Everyplace®, Oracle, Informix® (Retail/Distribution) Sybase (FSS), Rational®
Application Server	WebSphere® Application Server, Apache



<b>Infrastructure solutions</b>	<b>Vendor or solution</b>
Network and System Management	Tivoli®, VERITAS, BMC, Trustix, Compuware
Security and Firewall	Checkpoint, Symantec, CISCO, Trustix, SUSE Firewall, Tivoli
Clustering (High Availability)	Steeleye, VERITAS, PolyServe, Legato
Server Consolidation	VMware, Samba
Mail and Messaging	Sendmail, Bynari, Domino®, Caldera VMS, SUSE eMail Server
<b>e-business solutions</b>	
E-commerce	WebSphere Commerce Suite, eOneGroup
Accounting and CRM	ACCPAC, Sage
Business Intelligence and Data Warehouse	SAP BI, Brio, Crystal Decisions, Dimensional Insight, Hyperion
ERM/CRM/SCM (cross industry)	mySAP.com, QAD, BaaN Invensys
Retail Distribution	Relavis, MarCole, 360 Commerce, Caldera
Finance and Banking	Axiom, Sanchez, Rueters, Intelligent Markets, JD Edwards CRM for Financial Services, SUNGARD
Industrial (EDA)	Synopsis, Cadence, LSDC, Mentor Graphics

## 1.2 IBM @server BladeCenter and xSeries commitment to Linux

IBM is fully committed to the open source movement and believes that Linux will emerge as a key platform for e-business. That commitment is evident with the IBM @server xSeries and BladeCenter products. IBM @server xSeries and BladeCenter has a line of servers that support the Linux operating environment. These servers offer a variety of features and functions that separate them from

the competition. IBM is committed to Linux and open standards. For more information, visit the Web site:

<http://www-1.ibm.com/servers/eserver/linux/xseries/>

## 1.3 VERITAS commitment to Linux

VERITAS is also committed to Linux with a line of products supporting the OS platform. These products include:

- ▶ VERITAS Storage Foundation
- ▶ VERITAS Cluster Server
- ▶ VERITAS Volume Replicator
- ▶ VERITAS NetBackup
- ▶ VERITAS Storage Foundation for Database
- ▶ VERITAS Storage Foundation for Oracle RAC

## 1.4 VERITAS solutions

VERITAS has thousands of customers deploying cost-effective Linux solutions across their enterprise IT environments and has supported Linux software offerings for over five years. Today, VERITAS offers a full range of enterprise-ready storage management, availability, data protection, application performance management, and automation software on Linux. These VERITAS technologies make Linux an enterprise-ready and cost-effective alternative that meets the flexibility, availability, and interoperability requirements of multi-tier heterogeneous environments.

This section provides a brief summary of VERITAS solutions. For more complete information about these solutions, visit the VERITAS Web site:

<http://www.veritas.com>

### 1.4.1 VERITAS Storage Foundation for Linux

As data volumes grow, companies want to feel confident that they can manage their storage environments effectively through streamlined administration. They are looking for data management solutions that provide continuous availability of mission-critical data, while delivering optimal performance and efficient manageability.

VERITAS Storage Foundation combines the industry-leading VERITAS Volume Manager and VERITAS File System to provide a complete solution for online storage management. With VERITAS Storage Foundation, you can group

physical disks into logical volumes to improve disk utilization and to eliminate storage-related downtime. VERITAS has dramatically simplified the move to Linux with a breakthrough utility called the *Portable Data Container*. You can now migrate data to the Linux platform in minutes in three easy steps:

1. Convert the file system.
2. Extract the snapshot volume and deport it to the new disk group.
3. Use VERITAS Volume Manager to import the new disk group to the Linux environment.

VERITAS also offers VERITAS Storage Foundation Cluster File System, which extends VERITAS File System to enable data sharing on SAN-attached systems. In addition, VERITAS Storage Foundation for Oracle RAC on Linux is available, which provides an industry-leading framework for implementing robust, manageable, and scalable Real Application Clusters that enable administrators to increase the availability of Oracle databases.

## **1.4.2 VERITAS NetBackup for Linux**

VERITAS NetBackup delivers enterprise data protection for the largest UNIX, Windows®, Linux, and NetWare enterprise environments. Treating data availability as a utility, large enterprises can leverage dynamic disk-based backup and recovery. VERITAS NetBackup offers optional database agents, enabling online, nondisruptive backup of mission-critical applications.

## **1.4.3 VERITAS Volume Replicator for Linux**

VERITAS Volume Replicator, based on VERITAS Storage Foundation, provides enterprise-class disaster recovery for mission-critical environments. It replicates data reliably to remote locations over any IP network, which preserves business continuity for planned site migrations or when disaster or site failure threaten mission-critical environments.

VERITAS Volume Replicator is an open solution that can replicate data between storage arrays from all major vendors. It makes efficient use of network resources, replicating only data blocks that change. VERITAS Volume Replicator maintains data consistency by ensuring that replication at secondary sites does not lag beyond a preset limit in synchronous or asynchronous mode. It scales up to 32 secondary replication sites at any location worldwide. Advisor Tool, included in VERITAS Volume Replicator, enables administrators to determine bandwidth needs before implementing a replication.

## 1.4.4 VERITAS Cluster Server for Linux

VERITAS Cluster Server, reduces planned and unplanned downtime, facilitates server consolidation, and effectively manages a wide range of applications in heterogeneous environments. Expanded automation, features for testing production disaster recovery plans without disruption, and intelligent workload management allow cluster administrators to move beyond reactive recovery to proactive management of application availability.

## 1.4.5 VERITAS Opforce for Linux

VERITAS OpForce, the industry-leading server provisioning solution, helps companies overcome server management and provisioning challenges. It simplifies the process of building, operating, and optimizing server infrastructures.

## 1.4.6 VERITAS i3 for Linux

The increasing complexity of business applications and the growing demand for higher application service levels have made application performance management a necessity for many corporate data centers. Organizations can run their applications at peak performance levels by using VERITAS i3 to eliminate bottlenecks and fine-tune applications. VERITAS i3 helps analyze and improve the efficiency of key business applications, including BEA, Oracle, PeopleSoft, SAP, and Siebel, by detecting application performance problems, pinpointing their root cause, and recommending a solution. Depending on the chosen service levels, this process usually occurs before the user even notices problems.

VERITAS i3 has three main components:

- ▶ VERITAS Inform
- ▶ VERITAS Insight
- ▶ VERITAS Indepth

## 1.5 A world-class community of enterprise Linux partners

Top hardware and software companies are taking note of the surge in interest in enterprise Linux. Key VERITAS partners are moving ahead with solutions that make this reliable, secure operating system a viable option for the enterprise. VERITAS is partnering with the open-source community and enterprise Linux leaders such as IBM, HP, Intel®, Dell, Sun, Novell, and Red Hat to support

customers with a mature and robust Linux platform. In addition, VERITAS is returning kernel enhancements to the open-source community to help improve Linux scalability, performance, and reliability.





# UNIX to Linux migration considerations

When you migrate from UNIX to Linux, there are several factors to consider. This chapter describes those considerations generally.

## 2.1 Complexity of migrations

What is the essence of migration? In this section, we briefly discuss application and file migration and go further in depth in the sections that follow.

Application migration is the most complex work. It depends on the CPU architecture, programming language, its own design, application usage, and so on. Especially when the CPU architecture is different, you will need to consider recompiling the application and endianness (byte order) in some cases.

File migration differs with the type of file. The plain ASCII text file is inherently portable, but other files, such as binary, are not. To migrate an application binary file (executable) properly, you should recompile it for other operating systems.

File size is an important point of migration because the limitation of file size is different on each operating system. Currently, the Linux file size limitation is smaller than that of UNIX, potentially causing problems when migrating data.

Another technical detail is the inode information that the file system maintains. The inode information is used for creating files and directories, and the information is held by file system. Because files and directories cannot be created without inodes, you should consider the limitations of the maximum number of inodes. For more information see “The inode data structure” on page 11.

In brief, there are different types of migration that depend on migrated objects. Because migration objects may change the migration method, and migration requirements may change migration objects, migration is very complex work. Thus, you should consider the migration requirements very carefully.

## 2.2 Migration challenges

This section describes UNIX to Linux migration challenges.

### 2.2.1 Operating system dependence

The kernel is the core of an operating system. It provides the basic functions of the operating system, such as memory management, disk management, task management, and so on. Hardware management of the kernel depends on device drivers. The kernel uses them if it needs them. All functions of the operating system are maintained by the kernel. One view of the kernel is that it virtualizes the hardware resources for applications.



The kernel has many parameters, which are called *kernel parameters*. You can tune the kernel parameters, and the value of these parameters can effect the performance of the kernel. Rightly, the kernel parameter differs between Linux and Solaris because the design and the implementation differs between them. Thus, you should consider some kernel parameters when planning your data migration.

The most important consideration is the limitations and differences caused by an operating system. In many cases, you should consider the following items prior to migration.

### **File size**

The file size is a typical consideration of data migration, especially when you are considering a database migration. The file size limitation of the current shipping Linux kernel (2.4.x) is smaller than that of Solaris. The maximum possible file size in Linux is currently 2 TB. However, you should take care when moving beyond 2 GB because some utilities may not be large-file aware.

VERITAS File System on Linux is capable of creating sparse files, as large as 16 TB. However, there cannot be more than 2 TB of data in those files. The purpose of having a limit beyond 2 TB is that some applications that could be migrated to Linux may require a higher limit than 2 TB.

### **File system size**

The file system size is also an important consideration. Linux is currently limited to a maximum file system size of 2 TB.

For example, the limitation of a file system size may be a consideration of database migration. Because databases can use large files, file system size as well as file size can become a problem.

### **The inode data structure**

A data structure holding information about files in a UNIX file system is known as the *inode* or *index-node*. The inode is used for creating a file and directory in Linux and UNIX. Every file and directory in a file system is assigned an inode number. The inode contains at minimum of the following information:

- ▶ Permission (readable, writable, executable, and so on)
- ▶ File type
- ▶ File size
- ▶ Access time
- ▶ Modified or updated time
- ▶ File pointer table (which points to the place of the file)

This information is held by file system. Though there may be enough disk space, files or directories cannot be created if the inodes are used up. The maximum number of inodes differs between Linux and UNIX. However, the maximum number of inodes are tied to the file system being used and not the operating system.

VERITAS File System does not have any limitations on the number of inodes because they are dynamically allocated as files and directories are created.

## **Directory hierarchy**

The operating system manages files using the directory hierarchy which has a tree structure. For example, the password file is placed in the `/etc/passwd` directory on UNIX and Linux. Although some files are put on a common place between UNIX and Linux, there are also files which are not.

The consideration of directory hierarchy is the matter of file placement and directory structure. It can be a problem of migrating an application which expects a specific directory name or file name. Directory hierarchy is different between Linux and UNIX. It is needed to modify the design of the application in some cases.

## **System libraries**

Some system libraries are placed in a specified directory, such as `/lib`. The most important library is `libc`, which provides basic functions of the C programming language.

System libraries can present problems when migrating an application. If there is no difference between the function name and its design, the migration is performed without a problem. If there is no function name which the application expects, or if the return value of the function differ by the same name between Linux and UNIX, you should modify the application for these differences.

It is possible that the library which the application expects does not exist. In this case, an alternative library that includes the function which the application expects is searched for and located.

Because applications need many system libraries, you should consider carefully the system libraries before migrating.

## **Header files**

Some header files are placed in a specified directory, such as `/usr/include`. Header files have many data items, data structures, and data type definitions. Data type definition is really important, because some data type differs between Linux and Solaris.

This matter is similar to system libraries. Because the header file which the application expects does not exist, the application is not migrated. If the header file exists, it is possible that it does not include a data item or data structure which the application expects.

You should consider the type definition of data or data structure for the header files. If the header file exists and the data or data structure also exists, it would be a problem if the data type definition is different.

## Commands

Some applications may require a command using a command script or system call. Because the placement of the command (path) is important, you should consider the command that an application uses.

It is possible the command which is expected by the application does not exist or is in different directory. In this case, you will need to modify the application to work correctly.

In addition, if there is a difference of commands between Linux and UNIX, you should verify carefully the return value and options of the command.

## 2.2.2 File management dependence

Because there is a difference between the file management of Linux and UNIX, you should consider file management dependence for data migration.

### ACL

The access control list (ACL) allows you to strictly manage files and directories. Traditional UNIX permissions are limited. They can control permissions on a file or directory using only *user*, *group* and *other*. Traditional UNIX permissions allows only one type of permission. ACL provides multiple types of permission for users.

Unfortunately, Linux 2.4 does not support ACL officially. If you need ACL as a user requirement, that would be a matter of concern for data migration.

**Note:** Linux 2.6 supports ACL, and it will be used for future Linux distribution.

### Quota

Quota allows users to specify limits on two aspects of disk storage:

- ▶ The number of inodes a user or a group of users may possess
- ▶ The number of disk blocks that may be allocated to a user or a group of users.

The configuration of quota is written in the operating system's configuration files. There is no guarantee that compatibility is held between Linux and UNIX. Quota is needed to make sure that the configuration functions properly before migrating data. For more information, see "Maintenance command: fscdsconv(1M)" on page 135.

### File ownership

All files and directories have a user ID and a group ID. They should belong to a specific user and a specific group. You can verify these IDs using the `ls -l` command. For example:

```
Solaris # ls -l /etc/passwd
-r--r--r--  1 root    sys          616 Sep  1 15:10 /etc/passwd
```

In this example, `/etc/passwd` has `root` as the user ID and `sys` as group ID. However, the group ID of `/etc/passwd` differs between Linux and UNIX. On Linux, the group ID of `/etc/passwd` is `root`.

```
Linux # ls -la /etc/passwd
-rw-r--r--  1 root    root          2198 Aug 30 16:26 /etc/passwd
```

If an application expects the same group ID of `/etc/passwd`, it will not work correctly. In addition, a well known user ID and group ID are common between UNIX and Linux, but it is possible that their own ID value may differ. For example, the value of `bin` group is 2 on Solaris, but the value of `bin` group is 1 on Linux.

You should consider file ownership for data migration and application migration to be sure that the user ID and the group ID are set for the value not the name.

## 2.2.3 Endianness

The sort of byte ordering is called *endianness*. There are two types of endianness: *big endian* and *little endian*. Big endian means the most significant byte (MSB) comes first, and little endian means the least significant byte (LSB) comes first.

RISC architecture such as POWER™ or SPARC has big endian, and Intel architecture has little endian. The difference of endianness has a large impact in case of using a value more than 2 bytes in length. If endianness is different, the migrated application should take care of endianness.

For example, when 4 bytes value 0x12345678 is stored in big endian, MSB is 0x12 and LSB is 0x78. However, the value is changed as 0x78563412 in little endian because LSB comes first. If there is a big difference between these two values, the result completely differs. So, you should consider the data that depends on endianness.

## 2.2.4 Different techniques of migrating data

In this section, we discuss the different techniques of migrating data.

### Network

The basic technique for migrating data is using the network. The easiest way is using RCP (remote copy) or SCP (secure remote copy). This configuration is simple and better for small data transfer. FTP is better for large data transfer or for transferring a large number of files at one time. Because NFS is able to think as file system, the user does not need to be concerned with the network.

However, techniques using the network can present issues. The performance of data transfer depends on the network bandwidth and network topology. In general, these techniques are not suited for migrating lots of data, because transferring lots of data takes a long time.

### Replication

How about replication using array subsystem? Of course, the performance is much better than using the network. This method is really effective, but it is expensive. Because replication needs to use of a huge disk space, the cost would be increased according to disk space growth.

Replication using software is rather less expensive than using hardware, but its performance would be worse than using hardware. When thinking of the cost per transfer rate, it is difficult to decide which is the better method.

### Backup and restore

Migration using backup is good way to support a huge amount of data. In this case, tape library is often used because it is suited for safekeeping and delivering huge data. However, this method is not suited for migrating data because of its performance. Because it takes a long time for backup and restore, the cost of this method is considered high.

### Metadata

Metadata is called data about data. It is not the data itself. Metadata describes the structure of data and how the data was collected. For example, simple table structured data such as a phone book has several kind of items. The items such as last name and phone number are metadata, while the actual last name and phone number are *not*. Metadata is the key to understanding the structure of data.

Metadata describes generally the attribution of data such as data type, data length, and data structure. It also describes the structure of whole data such as database design. By using metadata, the actual data can be considered as

abstract data. Because abstract data has highly portability, data migration does not depend on destination of migration.

Thus, metadata is useful for migrating data. Since the data structure is known beforehand, it becomes comparatively easy to rebuild data after migration. If data is migrated without metadata, you will need to convert data into a new format, and converting data would take much longer.

Migration using reconnection may be the best method for migrating data. If it is possible, the time and cost for performing data migration are dramatically decreased because the data does not move. Migration using reconnection is realized by Portable Data Containers provided by VERITAS Storage Foundation 4.0. Because Portable Data Containers provide file system conversion using metadata, data migration which does not depend on an operating system is possible without moving data.



# VERITAS Storage Foundation

This chapter discusses the key features of VERITAS Storage Foundation 4.0 as well as the concepts of storage management.

## 3.1 VERITAS Storage Foundation overview

VERITAS Storage Foundation combines VERITAS Volume Manager and VERITAS File System to provide a solution for online storage management. With VERITAS Storage Foundation, physical disks are grouped into logical volumes to improve disk utilization and reduce wasted space (also known as *white space*) that is very common in storage configurations. In addition, VERITAS Storage Foundation gives administrators the flexibility to move data between different operating systems and storage arrays, balance I/O across multiple controllers to improve performance, create multiple volume file systems, dedicate volumes for certain types of files or file system structures, and other features that are required for enterprise storage solutions.

With VERITAS Storage Foundation, coupled with IBM @server xSeries servers, you can design real enterprise solutions. IBM @server, designed with scalability, reliability, and security in mind, enables VERITAS Storage Foundation to make the absolute best use of system resources.

VERITAS Storage Foundation consists of two core products; VERITAS Volume Manager and VERITAS File System. While both products are capable of operating independently of each other, key features of Storage Foundation are only available when the two products are used together. Thus, they are sold and treated as one product. A Java™-based graphical user interface (VERITAS Enterprise Administrator) is also bundled with VERITAS Storage Foundation, enabling the administrator to manage the storage solution from anywhere in the enterprise.

An additional benefit of using both products together is automatic configuration and performance tuning which results in increased performance and reduced management effort.

### VERITAS Volume Manager

VERITAS Volume Manager enables virtualization of the storage, allowing system and storage administrators to work with logical names rather than physical device addresses. This virtualization simplifies the management effort, reduces the chance for errors, and reduces cost because less time is spent managing the storage.

VERITAS Volume Manager includes features common to a volume manager such as frequently used RAID levels (0, 1, 1+0, 0+1, and 5) as well as advanced technologies such as policy-based Dynamic Multi Pathing and Instant Snapshots. For more information about these technologies, see 3.4, “Dynamic Multi Pathing” on page 21 and 3.6, “FlashSnap” on page 23.



## **VERITAS File System**

VERITAS File System is an enterprise journaling file system designed for performance and reliability. However, it is capable of very advanced features. With numerous tuning options to tailor the file system for almost any environment and workload, it is possible to achieve higher performance than any other file system. When used with VERITAS Volume Manager, features such as Quality of Storage Services and Portable Data Containers are enabled, bringing the next generation enterprise level functionality to Linux.

The remaining sections explain the key features of Storage Foundation 4.0.

## **3.2 Quality of Storage Services**

Quality of Storage Services is a new feature of Storage Foundation 4.0. From a very high level, it can be viewed as a disk-based Hierarchical Storage Management (HSM) solution. However, it is much more advanced than your normal HSM solution.

### **Understanding HSM**

A common situation in data centers is that there is not an infinite amount of disk storage available. This type of situation was especially apparent 20 years ago when disks were very expensive and tape based storage was a cheaper alternative. A technology was developed to move files from disks onto tape. However, rather than deleting the file on disk after it was moved, it created a stub file on disk. When the user then accessed the stub file, it would be recalled from tape to disk. In this technology, a policy based framework was used to determine when a file would be eligible to be moved from disk to tape. For example, a rule might be:

This file has not been used for 30 days, move it to tape to free up space in the file system.

Because tape technologies are slow, HSM is considered near useless in many environments where the user or application has high performance expectations and does not have time to wait for the tape to reboot, to load the correct tape, and to restore the file.

### **VERITAS Quality of Storage Services**

VERITAS Quality of Storage Services addresses one of the same issues as HSM, to lower the cost of storing data. However, it does so without sacrificing performance. It is actually more likely to improve performance than to reduce performance.

VERITAS Quality of Storage Services relies on a new concept in Volume Manager 4.0, called *volume set* (see Figure 3-1). A volume set is a collection of volumes in Volume Manager that are grouped together. A volume set is different from a volume because a volume typically contains one or several disks or Logical Unit Numbers (LUNs) rather than volumes as in a volume set. What makes a volume set so powerful is that each of the volumes in a volume set can have different RAID characteristics, size, and physical disks. This flexibility radically changes the way we design and create volumes.

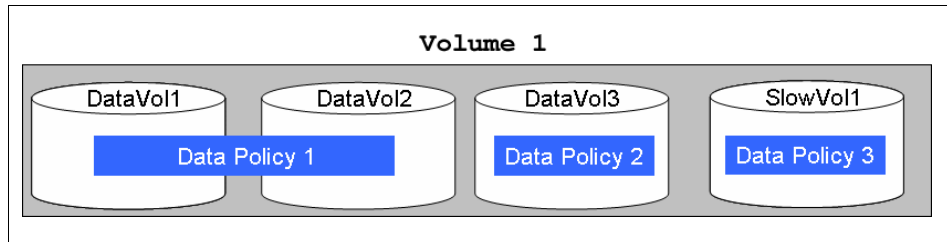


Figure 3-1 Example of a volume set with three policies overlaid

A volume set does not make sense unless the file system understands it and can use it. A policy engine in the file system does just that and with a simple-to-use graphical user interface (GUI). So, you can create and assign policies on a per sub directory granularity. The GUI is even capable of displaying what effect the policy would have before applying it, reducing the potential for mistakes.

A policy may look like this:

All files larger than 64KB and older than 10 days, move to volume 2.

Then there could be an additional policy:

All files larger than 64KB and older than 30 days, move to volume 3.

And an additional policy:

All files owned by user id 12345 (John Doe), excluding .pdf files, save these to volume 4.

You could continue to add policies in this manner.

The power of VERITAS Quality of Storage Service comes from the fact that it is invisible to the user or application. All movement of data happens at the block level and is abstracted by the file system.

Because the policies are extremely flexible and numerous different policies can be created, it is possible to create a storage hierarchy that enables higher performance than what would otherwise had been possible. If unused files are

moved to secondary volumes, the cache space that they potentially would have been allocated is left for the active files, giving a significant performance boost.

### 3.3 Multi Volume File System

What makes Quality of Storage Service possible is the VERITAS Multi Volume File System. It enables the file system to understand the concept of a volume set. When creating a file system or extending a file system, one or more volumes are specified, and the VERITAS File System uses the volumes as the policies dictate. No downtime is required to configure or enable this technology.

There are three types of data in a file system;

- ▶ **Metadata.** Stores structural information for the data files as well as other file system information. For example, metadata contains file size and file owner. Metadata mostly uses a random access pattern.
- ▶ **Log.** Stores every critical operation that is done on the file system. If the system crashes, the log ensures that the critical operations are replayed, removing the need of a file system check, which can take a significant amount of time (hours in the event of large file systems). The log typically uses sequential access pattern.
- ▶ **Data.** The user data. It can use either a random or sequential access pattern.

These three different types of data are usually accessed in different performance patterns, forcing the file system or operating system to make trade-offs on how to optimize the performance.

VERITAS Multi Volume File System is designed to remove any trade-offs required to maximum performance. Multi Volume File System technology allows the file system to separate out the different types of data to separate volumes. A volume, as part of a volume set, with the appropriate size and characteristics can be dedicated as metadata or log volume while other volumes handle the data. This design can significantly improve overall performance.

### 3.4 Dynamic Multi Pathing

A challenge with storage area networks (SAN) and powerful IBM @server environments is providing the server with enough bandwidth from the storage. One or even two Fibre Channel connections can easily be saturated in today's compute-intensive environments. Even worse, hardware problems may stop the server from processing data creating unnecessary downtime. Either of these problems can be mitigated with the use of multi pathing software.

The challenge with multi pathing software is that it needs to be intelligent enough to understand the hardware attached to the server and also the I/O patterns coming from the application. In addition, it needs to handle the unpredictable behavior of a storage area network.

VERITAS Volume Manager Dynamic Multi Pathing addresses all of these concerns. VERITAS has extensive test labs that test storage arrays from various vendors, including IBM FASTt storage. Because of its popularity and extensive functionality, VERITAS and IBM developed an Array Support Module for the FASTt storage series. The Array Support Module helps VERITAS optimize the data flow to the storage array as well as understand other properties of the array.

In addition to understanding an array better with the Array Support Module, Volume Manager provides a policy based multi pathing engine. Various policies are available for different computing environments and SAN topologies, making the storage more available and squeezing more performance out of the SAN.

## 3.5 Intelligent Storage Provisioning

Many storage administrators and architects will recognize the situation where a contractor has designed a storage solution and a year later a file system needs to be resized. However, the contractor is long gone, and insufficient documentation left behind is not of much help. In a situation like this, it would be nice to have help from the system itself rather than spending time trying to figure out what the storage configuration appears to be.

VERITAS Intelligent Storage Provisioning is designed to help reduce the amount of time spent on managing storage. The goal is to automate repetitive tasks and to reduce the chance for errors. Another benefit of using VERITAS Intelligent Storage Provisioning is that the storage architect's design of volumes are kept with the storage, no matter which server it is attached to. This feature ensures that the design intent is adhered to throughout the entire life cycle of the volume.

With VERITAS Intelligent Storage Provisioning, it is possible for the storage architect to specify templates for storage pools rather than just creating volumes. This design changes the concept of storage management from device management to a more abstract view where the application requirements are specified in a human language instead of hard to understand command line arguments.

The following is an example of a DB2 high-end template:

Volumes should always use IBM Shark storage that is of RAID-1 configuration and LUN's should be striped (already mirrored in hardware RAID) with a stripe-unit size of 128 KB and at least three columns wide.

To make the system even more self-sufficient, add the following to the template:

```
When reaching 90%, autogrow with 10 GB.
```

Storage templates can be made quite complex or kept simple. Either way, they are designed to reduce the management overhead of the storage.

## 3.6 FlashSnap

A challenge in today's data centers is keeping applications running all the time. Users expect to have everything online at all times. It is more difficult, for example, to do backups or to run nightly reports without adversely affecting system performance.

One solution that is frequently used is a point-in-time copy technology that allows the administrator to make a complete copy of the original data while the application is up and running. After the complete copy is finished, it can then be accessed over the SAN by another node.

You can use the point-in-time copy technology at different layers in the computing environment. You can apply it on the array level, creating copies of the entire LUN, or on the host, creating copies of volumes. Both of these technologies have their benefits, and some applications are better suited for one rather than the other.

VERITAS FlashSnap is a block-based point-in-time copy technology that is integrated into the VERITAS Volume Manager. With this technology, it is possible to make point-in-time copies of any storage being managed by Volume Manager. You can then export the snapshots over the SAN to another host running Volume Manager. You can accomplish all of this without reboots and without shutting down any applications.

FlashSnap 4.0 also has the capability of taking Instant Snapshots. Instant Snapshots are slightly different from the traditional snapshots. In the past, the user was forced to wait for the completion of the snapshot before being able to use it. With Instant Snapshots, the image is available instantaneously.

In some situations, it may not be desired to export the snapshot over the SAN. It may then be too expensive to take a full snapshot of the original volume because it will take up as much space as the original volume. FlashSnap has the capability of taking Space-Optimized Snapshots, allowing you to specify a snapshot volume size (for example, 10% of original volume). The snapshot uses that until it runs out of space. If the changes exceed 10%, then Volume Manager automatically expands the volume to the required size.

## 3.7 Portable Data Containers

Portable Data Containers are a combination of functionality in the VERITAS Volume Manager and VERITAS File System. They enable the administrator to move file systems and volumes between platforms with minimal effort. In short, Portable Data Containers makes data migration a very simple task.

Chapter 4, “VERITAS Portable Data Containers for migration” on page 25 discusses the details of Portable Data Containers.



# VERITAS Portable Data Containers for migration

This chapter discusses the details about VERITAS Portable Data Containers and presents a step-by-step migration example from Solaris to Linux.

## 4.1 The technical barriers that Portable Data Containers overcome

For data sharing across operating systems and hardware architectures to be possible, several requirements have to be met. These requirements include the SCSI layer and disk label. However, the device level, volume level, file system level, and application level are all required to be platform agnostic.

### 4.1.1 Operating system

This section discusses the SCSI layer and disk label.

#### SCSI layer

There are basic SCSI requirements for cross-platform data sharing to be possible. However, while they are generally not a problem in today's modern disk drives, you should consider the following requirements if you are migrating from an older environment to Linux:

- ▶ The disks must support SCSI mode-sense commands.
- ▶ EFI disks are not supported.
- ▶ The disk being migrated cannot be a boot disk
- ▶ You must configure the entire disk, not just a partition of the disk, for cross-platform data sharing.

#### Disk label

The first challenge to overcome when dealing with multiple platforms is the disk label. The disk label is stored in the first sector of the disk. If the operating system cannot understand the disk label, it cannot read the contents of the disk and is unable to use the disk until a new partition table has been written to the disk. Writing a new disk label to disk destroys the data on disk.

For Linux and Solaris, the platform block at sector 0 correctly identifies the disk for both operating systems.

For HP-UX and AIX®, we took a slightly different approach because of the logical volume managers on these systems. All CDS-enabled disks contain a AIX and HP-UX co-existence label that identifies the disk as being used by VERITAS Volume Manager. (See 4.2.2, “VERITAS Storage Foundation 4.0 installation and configuration” on page 33 for more information.)



## 4.1.2 Volume Manager

When VERITAS Volume Manager initializes a disk (for example, when it takes ownership of the disk the first time), it modifies the partition table on the disk. This modification is done so that Volume Manager can store configuration data on the disks. Storing the configuration database on the data disks makes the configuration portable between platforms, compared to using local disks for configuration data.

After disk initialization, the disk partition table is modified to contain two slices: a public region and a private region.

- ▶ The public region is the user available space of the disk (minus the private region). The public region is allocated for volumes created in the system.
- ▶ The private region contains configuration information, state information, and the configuration database intent log. This part of the disk is never modified directly by a user but only through the Volume Manager interface (GUI or command line interface). The private region generally uses only one cylinder of the disk, thus the space requirements are very small.

### Platform block size differences

The block size of an operating system is a multiple of the sector size (a minimum of 512 bytes). Linux and the different UNIX platforms have different block sizes. The block size is critical to reliably writing data and reading it on another platform. If the block size difference is ignored, then moving data may not work in all cases. For example, a different block size results in block boundary issues, and the system is not able to find the start of the data.

The block size directly affects the data written into the private region because it is written on a block boundary. There is an internal mechanism in Volume Manager that handles the block size difference for private region, called *VxVM ID block*. The user does not need to be concerned about the VxVM ID block.

To overcome the block size differences, a new Volume Manager disk format was introduced, called *cdsdisk*.

### Cross-platform data sharing disk format

The *cdsdisk* disk type is the default choice when initializing a disk in VERITAS Storage Foundation 4.0. The only limitation is that older versions of Volume Manager cannot understand this format.

It is required that all servers that may access a disk initialized with the *cdsdisk* format are running Storage Foundation 4.0 or later. If you are migrating from an older installation of Volume Manager, then the diskgroup (and disks) needs to be upgraded to the latest version for cross-platform data sharing to be possible.

Upgrading the on-disk format is an online operation and can be done while the system is running, reducing the downtime of your IBM environment.

### Platform device restrictions

Because Linux has a limitation with regards to the number of disks the operating system can support (or to be more specific, any device entry that requires a minor number), a new tunable parameter was added to the cdsdisk format. To prevent a storage administrator from creating a configuration that would be unable to migrate to Linux, a device quota can be set on the disk group. An error is displayed if the administrator is trying to go beyond the set limit. By default, this value is not set. If you have more than 254 disks to migrate, this parameter is useful.

### 4.1.3 File system

The file system needs to be capable of reading its data and metadata information about all supported architectures. VERITAS File System can read data written on any supported platform. However, it is not capable of converting the data because of the way that the data is written to the file system. An application writes a stream of bytes into the file system. It is not possible for the file system to know whether the application writes the data in an endian-dependent format.

The metadata is directly controlled by the file system and written in an endian-dependent format. Thus, the metadata written on a UNIX platform cannot be read on a Linux platform (small endian) without a conversion process.

**Note:** Moving a file system between UNIX platforms (Sun, HP and AIX) does not require any metadata conversion. An unmount or mount procedure is all that is required.

### Conversion process

The metadata conversion process is only required when moving to or from a Linux platform. A single command executes an offline conversion of the file system. The process is done in three steps:

1. Reads all the metadata. Makes sure that the file system is capable of supporting the destination platform. Because there are quite a few platform differences, it is possible that a file system may contain metadata that is not supported on another platform. For example, Linux only allows user IDs up to 65 536, while other UNIX platforms have limits up to 2 000 000 000.
2. Saves the metadata to a recovery file. The purpose for this file is to undo the conversion in case it fails due to unexpected system failure (for example,

power outage). There is more information about the usage of the recovery file in Chapter 6, “Best practices” on page 79.

3. Writes the converted metadata to the file system.

After step 3, the file system is ready to be moved. Disconnect the disks and connect them to the new server. Let the new server detect the disks, and the diskgroup can be imported and file system mounted.

#### 4.1.4 Applications

The file system and volume manager makes sure that the volumes and file system are portable between platforms. However, it is up to the application to understand the data format.

For example, a text file is easily portable between any platform as well as MP3 files and word processing documents. The only potential problem applications are databases and proprietary applications that write their data in a platform (endian) dependent format.

#### 4.1.5 Problems that Portable Data Containers do not address

The following are aspects with cross-platform data sharing that Portable Data Containers do not address:

- ▶ Portable Data Containers do not support concurrent sharing of data between platforms (for example, mounting a file system on more than one platform at the same time).
- ▶ Portable Data Containers do not convert the data in a file. As discussed previously, the application is responsible for writing the data and being able to read it on the new platform. Portable Data Containers make sure that the file system is mountable on the destination platform.
- ▶ File system quotas are not converted between platforms. You have to recalculate the quotas on the destination platform.
- ▶ Microsoft® Windows is not a supported target platform. While VERITAS has a Volume Manager product for Windows, a file system is not available, making functionality with Portable Data Containers difficult to achieve.
- ▶ Access control lists are not currently supported in VERTAS File System for Linux. These lists are not converted during the conversion process. However, they are not removed from the file system metadata. If the file system is moved back to the originating platform, metadata that is not used on the destination platform is still available.

## 4.2 Step-by-step migration example

This section provides a step-by-step migration example that we used in our testing lab.

### 4.2.1 Lab environment

Figure 4-1 provides an outline of the lab environment where we tested our migration procedures.

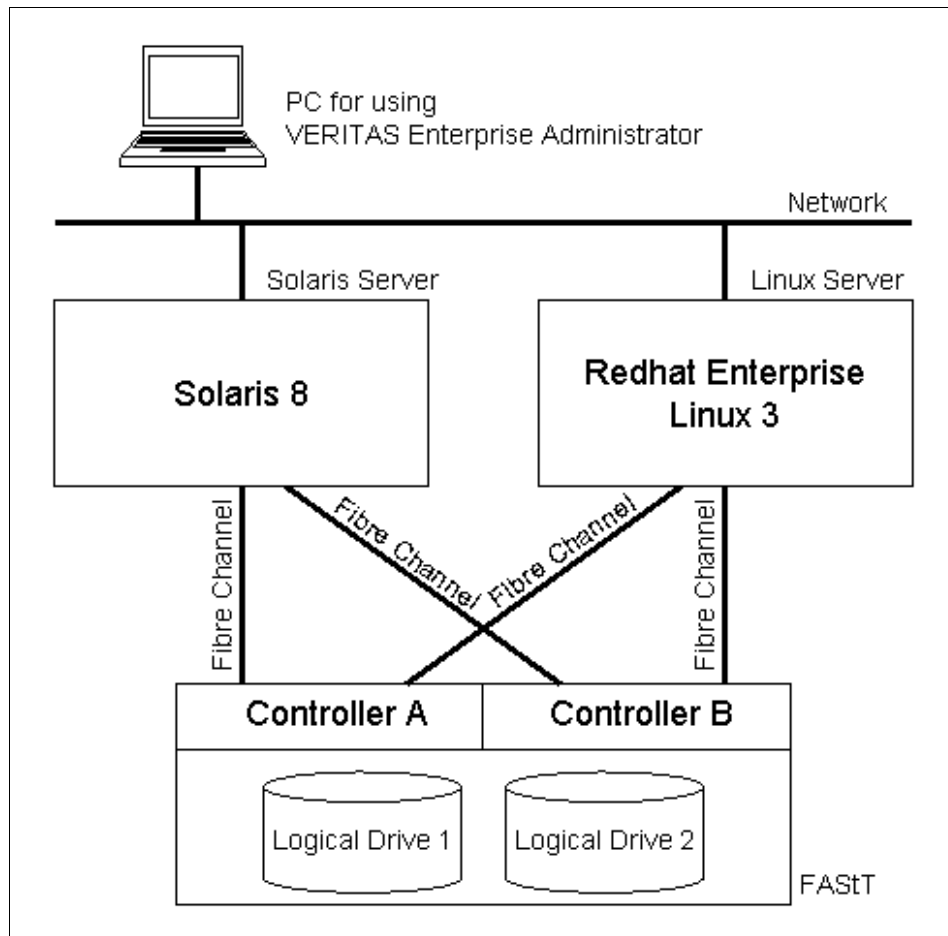


Figure 4-1 Lab environment

## **Linux server**

Our Linux server included the following features:

- ▶ Model: BladeCenter HS20
- ▶ OS: Red Hat Enterprise Linux 3 update 2 (kernel 2.4.21-15 SMP)
- ▶ CPU: Intel Xeon 2.0 GHz
- ▶ Memory: 2 560 MB
- ▶ Disk: 40 GB

## **Solaris server**

Our Solaris server included the following features:

- ▶ Model: Sun Ultra-60
- ▶ OS: Solaris 8 (5.8 Generic\_117350-06)
- ▶ CPU: Dual UltraSPARC-II 450 MHz
- ▶ Memory: 512 MB
- ▶ Disk: 18 GB internal boot disk

## **FAStT Storage subsystem**

Our DS4300 storage subsystem (formerly known as FAStT600) included the following features:

- ▶ Model: FAStT600
- ▶ Appware: ver.05.40.03.02
- ▶ Bootware: ver.05.40.03.02
- ▶ Disk: 36 GB x 14
- ▶ RAID configuration: (2) RAID 5 logical drive, 200 GB usable space per LUN
- ▶ Storage partitioning feature installed for creating multiple partitions
- ▶ Direct connection using fibre channel cable without fibre channel switch

## **QLogic Fibre Channel device driver**

Our Fibre Channel adapters used the following drivers:

- ▶ Solaris: QLA2300 rev.4.13.01 (non-failover, provided by SUN)
- ▶ Linux: QLA2300 version 7.00.61 (non-failover, provided by IBM)

## **Software**

The software products used for our configuration were:

- ▶ Management PC (Windows 2000)
- ▶ FAStT Storage Manager 8.4 (installed on the management PC)
- ▶ VERITAS Enterprise Administrator for Windows (installed on the management PC)

- ▶ VERITAS Array Support Library for FASTt for using Dynamic Multi Pathing, provided by VERITAS (downloaded from VERITAS support site, installed on both the Solaris and Linux servers)

## FASTt configuration

The UNIX to Linux migration using VERITAS Storage Foundation 4.0 Portable Data Containers requires a SAN connection of storage subsystems. Figure 4-2 shows an example of the FASTt drive mapping configuration:

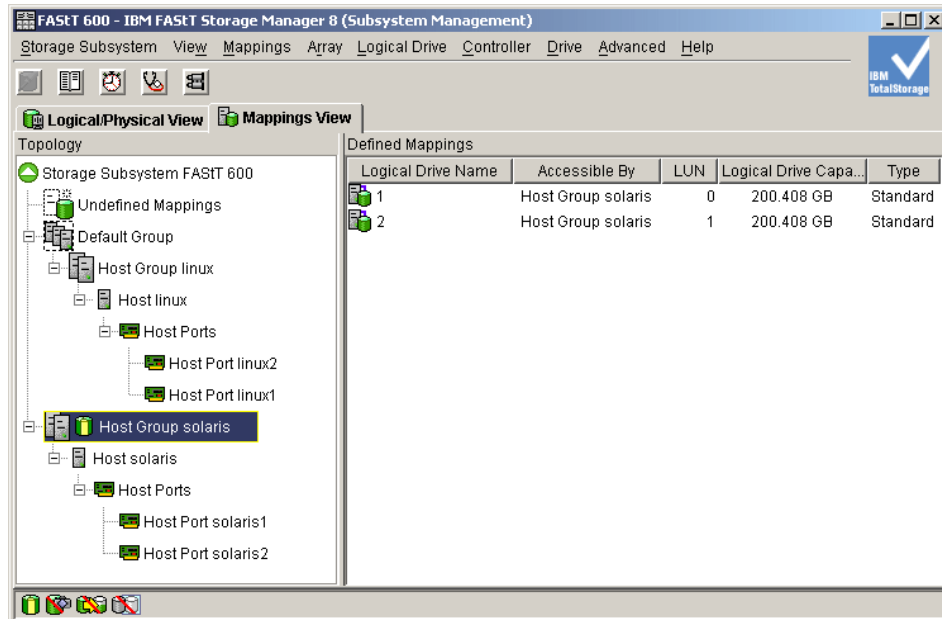


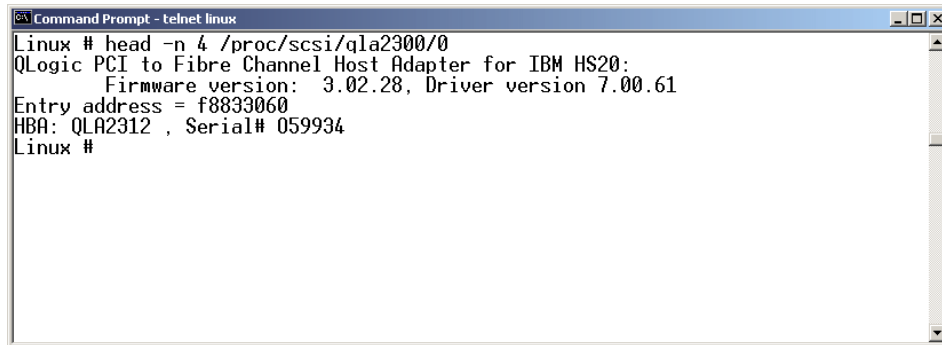
Figure 4-2 Initial logical drive mapping

In Figure 4-2, all logical drives are mapped to Solaris, and nothing is mapped to Linux. This configuration requires the storage partitioning feature. This feature should be installed to create multiple mappings of logical drives and the host bus adapter.

**Note:** We installed the IBM FASTt Storage Manager 8 (Subsystem Management) window shown in Figure 4-2 on the Management PC running Windows 2000.

On our Linux server, the Fibre Channel host bus adapter required a non-failover device driver for using Dynamic Multi Pathing (DMP). DMP is part of the VERITAS Volume Manager. DMP manages the access path of storage subsystems such as the FASTt.

In Figure 4-3, we used telnet to connect to the Linux server (the target system). Within the command prompt, we ran the **head** command to verify that the driver version did not have the string fo. If the driver version has the string fo, it is a failover driver.



```
Command Prompt - telnet linux
Linux # head -n 4 /proc/scsi/qla2300/0
QLogic PCI to Fibre Channel Host Adapter for IBM HS20:
    Firmware version: 3.02.28, Driver version 7.00.61
Entry address = f8833060
HBA: QLA2312 , Serial# 059934
Linux #
```

Figure 4-3 Verifying device driver of host bus adapter

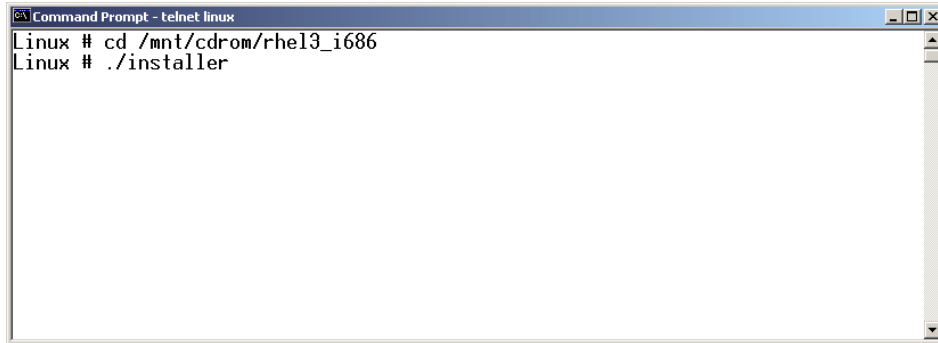
## 4.2.2 VERITAS Storage Foundation 4.0 installation and configuration

This installation of VERITAS Storage Foundation 4.0 is based on the description of installation manual. See *VERITAS Storage Foundation 4.0 Installation Guide* on the product CD for more details.

To install VERITAS Storage Foundation 4.0, simply run the installation shell script as shown in Figure 4-4 on page 34.

**Note:** In this example, we illustrate the installation of VERITAS Storage Foundation 4.0 on a Linux server. We also installed the VERITAS Storage Foundation 4.0 on a Solaris during. The installations were very similar.

1. Insert the VERITAS Software Storage Solutions 4.0 for Linux CD into your CD-ROM drive.  
For Solaris, insert the VERITAS Software Storage Solutions 4.0 CD 1 for Solaris.



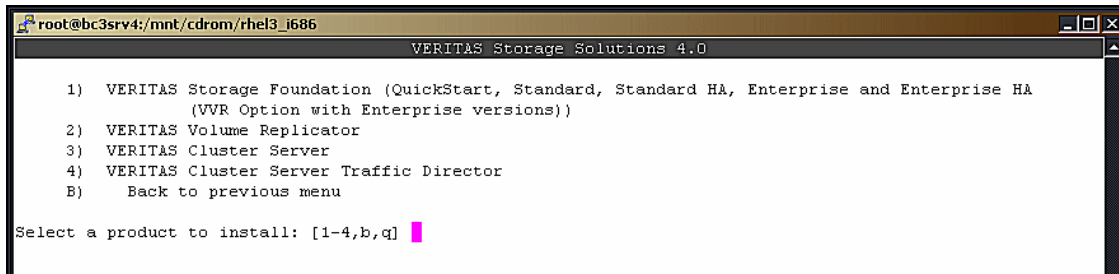
```
Command Prompt - telnet linux
Linux # cd /mnt/cdrom/rhel3_i686
Linux # ./installer
```

Figure 4-4 Beginning of installation

2. Telnet to the Linux server or open a command prompt at the Linux server and enter the following commands:

```
cd /mnt/cdrom/rhel3_i686
./installer
```

You will see a window similar to Figure 4-5.



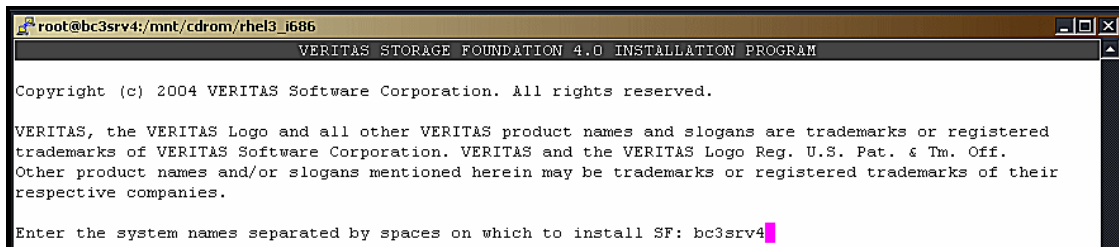
```
root@bc3srv4:/mnt/cdrom/rhel3_i686
VERITAS Storage Solutions 4.0

1) VERITAS Storage Foundation (QuickStart, Standard, Standard HA, Enterprise and Enterprise HA
   (VVR Option with Enterprise versions))
2) VERITAS Volume Replicator
3) VERITAS Cluster Server
4) VERITAS Cluster Server Traffic Director
B) Back to previous menu

Select a product to install: [1-4,b,q] █
```

Figure 4-5 VERITAS Storage Foundation 4.0 product installation window

3. For this example, we selected option 1 and pressed Enter. You will see a window similar to Figure 4-6.



```
root@bc3srv4:/mnt/cdrom/rhel3_i686
VERITAS STORAGE FOUNDATION 4.0 INSTALLATION PROGRAM

Copyright (c) 2004 VERITAS Software Corporation. All rights reserved.

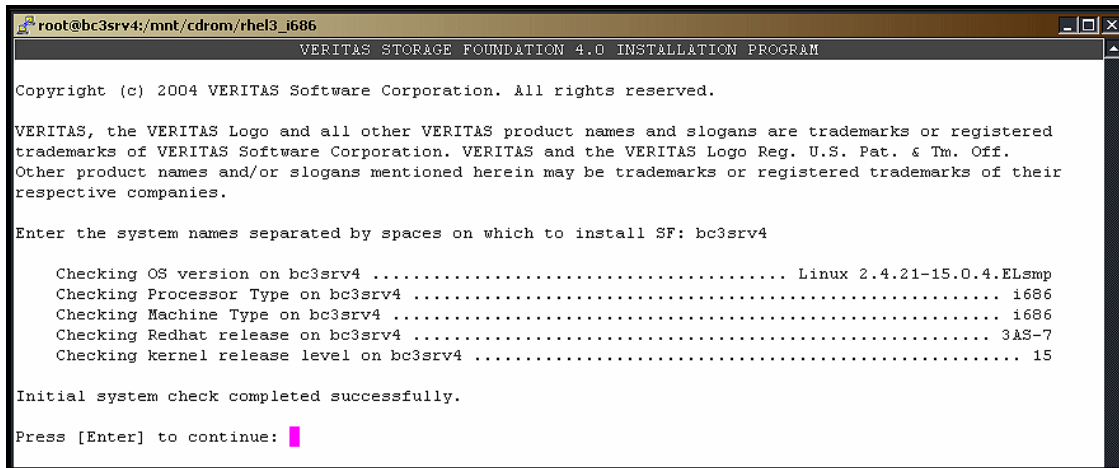
VERITAS, the VERITAS Logo and all other VERITAS product names and slogans are trademarks or registered
trademarks of VERITAS Software Corporation. VERITAS and the VERITAS Logo Reg. U.S. Pat. & Tm. Off.
Other product names and/or slogans mentioned herein may be trademarks or registered trademarks of their
respective companies.

Enter the system names separated by spaces on which to install SF: bc3srv4 █
```

Figure 4-6 VERITAS Storage Foundation 4.0 system names



4. Enter the system name on which VERITAS Storage Foundation 4.0 is to be installed. For this example, we typed bc3srv4. Press Enter. You will see a window similar to Figure 4-7.



```
root@bc3srv4:/mnt/cdrom/rhel3_i686
VERITAS STORAGE FOUNDATION 4.0 INSTALLATION PROGRAM

Copyright (c) 2004 VERITAS Software Corporation. All rights reserved.

VERITAS, the VERITAS Logo and all other VERITAS product names and slogans are trademarks or registered
trademarks of VERITAS Software Corporation. VERITAS and the VERITAS Logo Reg. U.S. Pat. & Tm. Off.
Other product names and/or slogans mentioned herein may be trademarks or registered trademarks of their
respective companies.

Enter the system names separated by spaces on which to install SF: bc3srv4

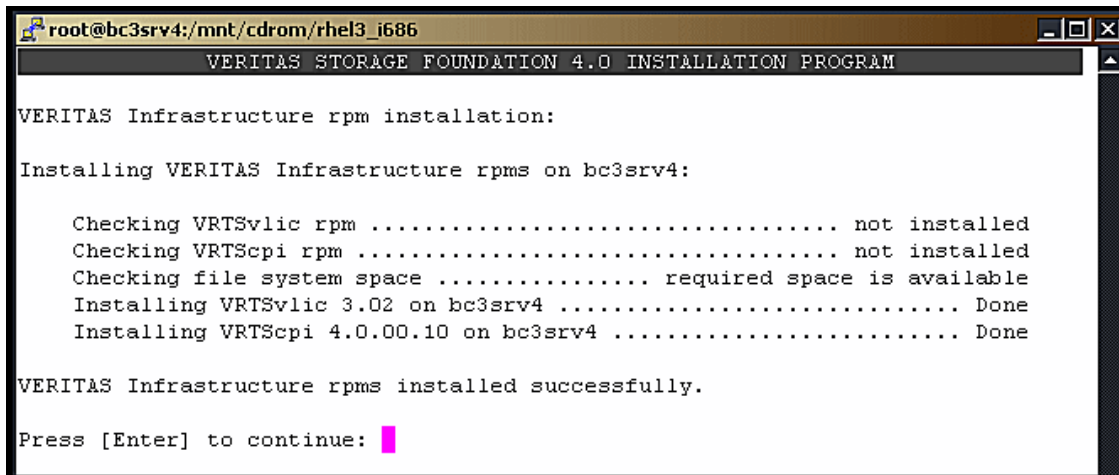
  Checking OS version on bc3srv4 ..... Linux 2.4.21-15.0.4.ELsmp
  Checking Processor Type on bc3srv4 ..... i686
  Checking Machine Type on bc3srv4 ..... i686
  Checking Redhat release on bc3srv4 ..... 3AS-7
  Checking kernel release level on bc3srv4 ..... 15

Initial system check completed successfully.

Press [Enter] to continue: █
```

Figure 4-7 VERITAS Storage Foundation 4.0 initial system check is executed

5. Press Enter to continue. You will see a window similar to Figure 4-8.



```
root@bc3srv4:/mnt/cdrom/rhel3_i686
VERITAS STORAGE FOUNDATION 4.0 INSTALLATION PROGRAM

VERITAS Infrastructure rpm installation:

Installing VERITAS Infrastructure rpms on bc3srv4:

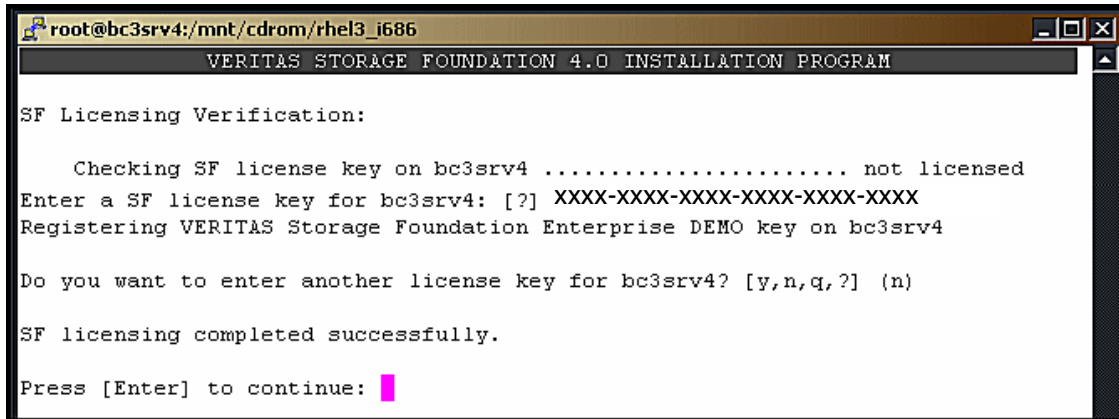
  Checking VRTSvlic rpm ..... not installed
  Checking VRTScpi rpm ..... not installed
  Checking file system space ..... required space is available
  Installing VRTSvlic 3.02 on bc3srv4 ..... Done
  Installing VRTScpi 4.0.00.10 on bc3srv4 ..... Done

VERITAS Infrastructure rpms installed successfully.

Press [Enter] to continue: █
```

Figure 4-8 VERITAS Storage Foundation 4.0 infrastructure rpm installation

6. After the infrastructure rpms are installed, press Enter. You will see a window similar to Figure 4-9.



```
root@bc3srv4:/mnt/cdrom/rhel3_i686
VERITAS STORAGE FOUNDATION 4.0 INSTALLATION PROGRAM

SF Licensing Verification:

    Checking SF license key on bc3srv4 ..... not licensed
Enter a SF license key for bc3srv4: [?] XXXX-XXXX-XXXX-XXXX-XXXX-XXXX
Registering VERITAS Storage Foundation Enterprise DEMO key on bc3srv4

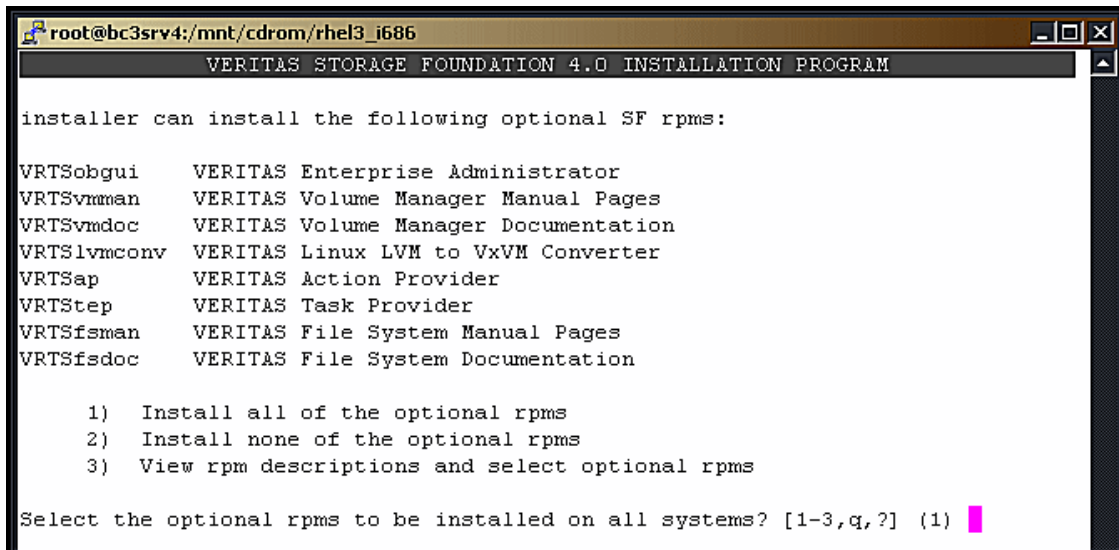
Do you want to enter another license key for bc3srv4? [y,n,q,?] (n)

SF licensing completed successfully.

Press [Enter] to continue: █
```

Figure 4-9 VERITAS Storage Foundation 4.0 Storage Foundation Licensing Verification

7. When asked to enter a Storage Foundation license key for the server, enter the license key that has been supplied with the software. When you have completed the Storage Foundation licensing, press Enter. You will see a window similar to Figure 4-10.



```
root@bc3srv4:/mnt/cdrom/rhel3_i686
VERITAS STORAGE FOUNDATION 4.0 INSTALLATION PROGRAM

installer can install the following optional SF rpms:

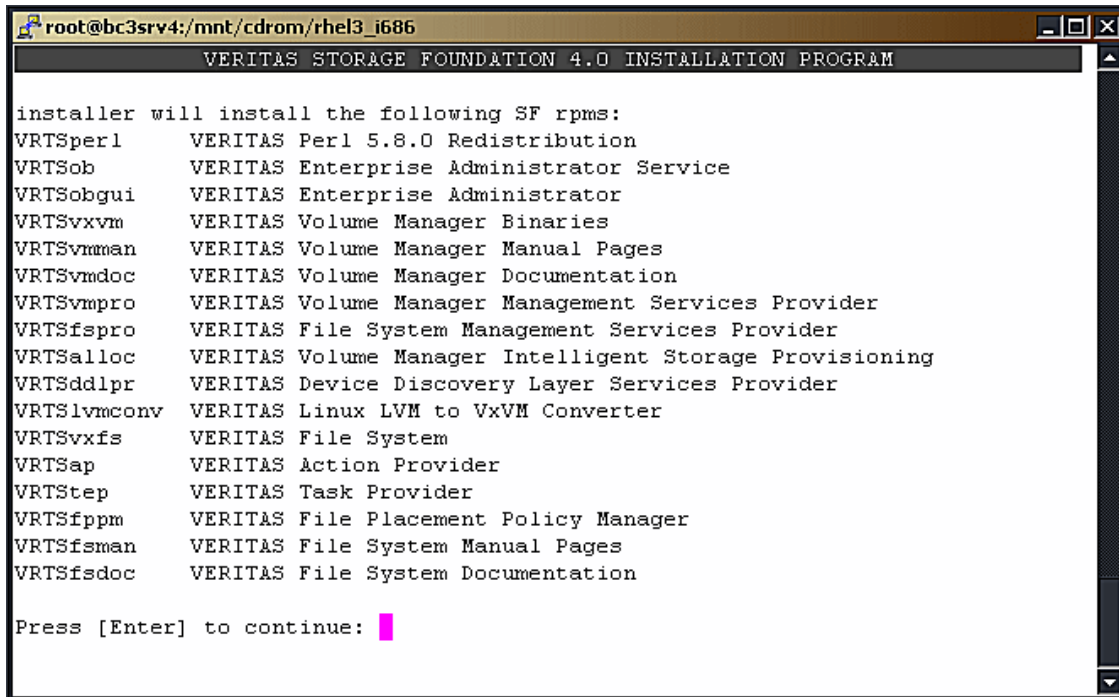
VRTSobgui   VERITAS Enterprise Administrator
VRTSvman    VERITAS Volume Manager Manual Pages
VRTSvmdoc   VERITAS Volume Manager Documentation
VRTSLvmconv VERITAS Linux LVM to VxVM Converter
VRTSap      VERITAS Action Provider
VRTStep     VERITAS Task Provider
VRTSfsman   VERITAS File System Manual Pages
VRTSfsdoc   VERITAS File System Documentation

    1) Install all of the optional rpms
    2) Install none of the optional rpms
    3) View rpm descriptions and select optional rpms

Select the optional rpms to be installed on all systems? [1-3,q,?] (1) █
```

Figure 4-10 VERITAS Storage Foundation 4.0 optional rpms

8. Make a selection to install all, some, or none of the optional rpms. In this example, we installed all the optional rpms. Press Enter. You will see a window similar to Figure 4-11.



```
root@bc3srv4:/mnt/cdrom/rhel3_i686
VERITAS STORAGE FOUNDATION 4.0 INSTALLATION PROGRAM

installer will install the following SF rpms:
VRTSperl      VERITAS Perl 5.8.0 Redistribution
VRTSob        VERITAS Enterprise Administrator Service
VRTSobgui     VERITAS Enterprise Administrator
VRTSvxvm      VERITAS Volume Manager Binaries
VRTSvman      VERITAS Volume Manager Manual Pages
VRTSvmdoc     VERITAS Volume Manager Documentation
VRTSvmpro     VERITAS Volume Manager Management Services Provider
VRTSfspro     VERITAS File System Management Services Provider
VRTSalloc     VERITAS Volume Manager Intelligent Storage Provisioning
VRTSddlpr     VERITAS Device Discovery Layer Services Provider
VRTSvvmconv   VERITAS Linux LVM to VxVM Converter
VRTSvxfs      VERITAS File System
VRTSap        VERITAS Action Provider
VRTStep       VERITAS Task Provider
VRTSfppm      VERITAS File Placement Policy Manager
VRTSfsman     VERITAS File System Manual Pages
VRTSfsdoc     VERITAS File System Documentation

Press [Enter] to continue: █
```

Figure 4-11 VERITAS Storage Foundation 4.0 installer

9. Press Enter to install the Storage Foundation rpms. You will see a window similar to Figure 4-12 on page 38.

```
root@bc3srv4:/mnt/cdrom/rhel3_i686
Checking VRTSperl rpm ..... not installed
Checking VRTSob rpm ..... not installed
Checking VRTSobgui rpm ..... not installed
Checking VRTSvxvm rpm ..... not installed
Checking VRTSvman rpm ..... not installed
Checking VRTSvmdoc rpm ..... not installed
Checking VRTSvmpro rpm ..... not installed
Checking VRTSfspro rpm ..... not installed
Checking VRTSalloc rpm ..... not installed
Checking VRTSddlpr rpm ..... not installed
Checking VRTSsvmconv rpm ..... not installed
Checking VRTSvxfs rpm ..... not installed
Checking VRTSap rpm ..... not installed
Checking VRTStep rpm ..... not installed
Checking VRTSfppm rpm ..... not installed
Checking VRTSfsman rpm ..... not installed
Checking VRTSfsdoc rpm ..... not installed
Checking file system space ..... required space is available
Checking for patch(1) rpm ..... version 2.5.4 installed
Checking vxsvc process ..... not running

Installation requirement checks completed successfully.

Press [Enter] to continue: █
```

Figure 4-12 VERITAS Storage Foundation 4.0 installation requirement checks

10. VERITAS Storage Foundation 4.0 checks the system to determine if any of the rpms have been installed. After the checks complete, press Enter. You will see a window similar to Figure 4-13.

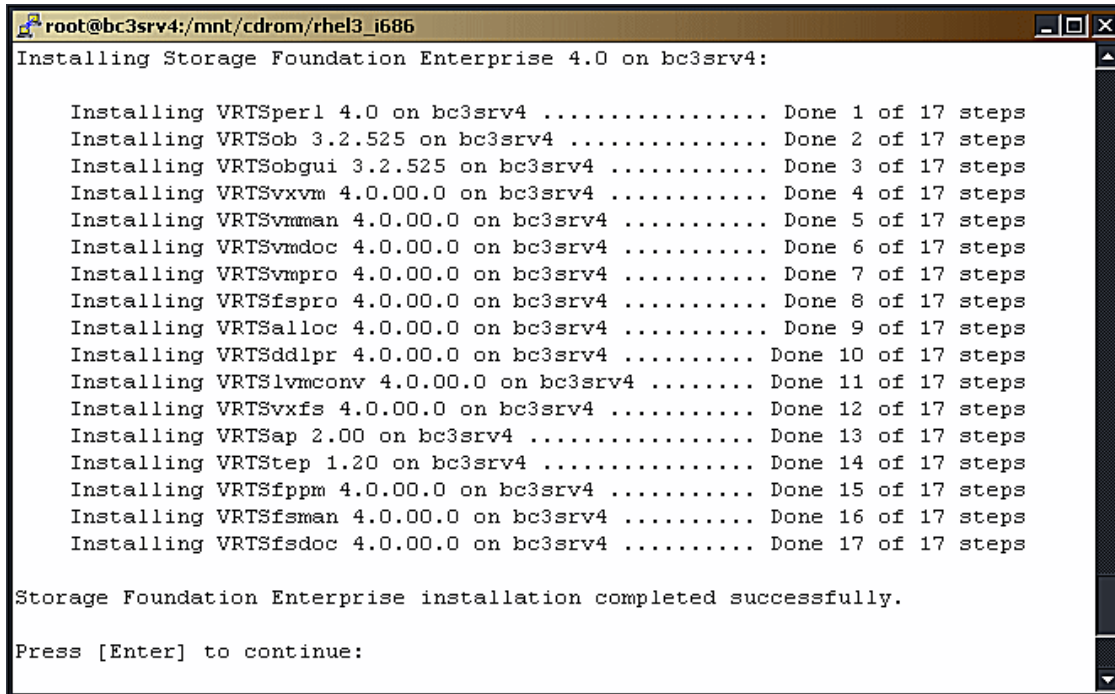
```
root@bc3srv4:/mnt/cdrom/rhel3_i686
VERITAS STORAGE FOUNDATION 4.0 INSTALLATION PROGRAM

It is possible to install SF rpms without performing configuration.

It is optional to configure SF now. If you choose to configure SF later, you
can either do so manually or run the installsf -configure command.
Are you ready to configure SF? [y,n,q] (y) █
```

Figure 4-13 VERITAS Storage Foundation 4.0 option

11. At this window, you are ready to configure VERITAS Storage Foundation 4.0. Type `y` and press Enter. You will see a window similar to Figure 4-14.



```
root@bc3srv4:/mnt/cdrom/rhel3_i686
Installing Storage Foundation Enterprise 4.0 on bc3srv4:

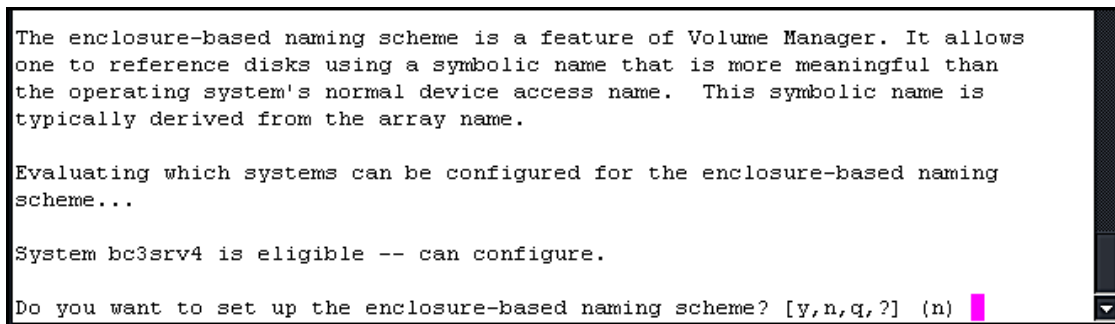
Installing VRTSperl 4.0 on bc3srv4 ..... Done 1 of 17 steps
Installing VRTSob 3.2.525 on bc3srv4 ..... Done 2 of 17 steps
Installing VRTSobgui 3.2.525 on bc3srv4 ..... Done 3 of 17 steps
Installing VRTSvxvm 4.0.00.0 on bc3srv4 ..... Done 4 of 17 steps
Installing VRTSvman 4.0.00.0 on bc3srv4 ..... Done 5 of 17 steps
Installing VRTSvmdoc 4.0.00.0 on bc3srv4 ..... Done 6 of 17 steps
Installing VRTSvmpro 4.0.00.0 on bc3srv4 ..... Done 7 of 17 steps
Installing VRTSfspro 4.0.00.0 on bc3srv4 ..... Done 8 of 17 steps
Installing VRTSalloc 4.0.00.0 on bc3srv4 ..... Done 9 of 17 steps
Installing VRTSddlpr 4.0.00.0 on bc3srv4 ..... Done 10 of 17 steps
Installing VRTSvmconv 4.0.00.0 on bc3srv4 ..... Done 11 of 17 steps
Installing VRTSvxfs 4.0.00.0 on bc3srv4 ..... Done 12 of 17 steps
Installing VRTSap 2.00 on bc3srv4 ..... Done 13 of 17 steps
Installing VRTStep 1.20 on bc3srv4 ..... Done 14 of 17 steps
Installing VRTSfppm 4.0.00.0 on bc3srv4 ..... Done 15 of 17 steps
Installing VRTSfsman 4.0.00.0 on bc3srv4 ..... Done 16 of 17 steps
Installing VRTSfsdoc 4.0.00.0 on bc3srv4 ..... Done 17 of 17 steps

Storage Foundation Enterprise installation completed successfully.

Press [Enter] to continue:
```

Figure 4-14 VERITAS Storage Foundation 4.0 successful install

12. After the rpms have been installed successfully, press Enter. You will see a window similar to Figure 4-15.



```
The enclosure-based naming scheme is a feature of Volume Manager. It allows
one to reference disks using a symbolic name that is more meaningful than
the operating system's normal device access name. This symbolic name is
typically derived from the array name.

Evaluating which systems can be configured for the enclosure-based naming
scheme...

System bc3srv4 is eligible -- can configure.

Do you want to set up the enclosure-based naming scheme? [y,n,q,?] (n)
```

Figure 4-15 VERITAS Storage Foundation 4.0 naming scheme

13. At this stage, you choose to set up the enclosure-based naming scheme. In this example, we typed n. Press Enter to continue. You will see a window similar to Figure 4-16.

```
root@bc3srv4:/mnt/cdrom/rhel3_i686
VERITAS STORAGE FOUNDATION 4.0 INSTALLATION PROGRAM

Evaluating which systems can now be started...

System bc3srv4 is eligible -- can be started.
Preparing to start VxVM on target systems...
Volume Manager

Note: The vxconfigd daemon will be started, which can take a while depending
upon the hardware configuration.

Begin initial start of VxVM on system bc3srv4
  Disabling enclosure-based naming ..... Done
  Starting vxconfigd for VxVM ..... Succeeded
Done with initial start of VxVM on system bc3srv4

Done with starting VxVM on target systems...

Press [Enter] to continue: █
```

Figure 4-16 VERITAS Storage Foundation 4.0 VxVM start

14. Allow Volume Manager to start. When complete, press Enter. You will see a window similar to Figure 4-17.

```
Many Volume Manager commands affect the contents or configuration of a disk
group. Such commands require that the user specify a disk group. This is
accomplished by using the -g option of a command or setting the
VXVM_DEFAULTDYG environment variable. An alternative to these two methods is
to configure the default disk group of a system.

Evaluating which systems can now have their default disk group configured...

System bc3srv4 is eligible -- can configure the default diskgroup.

Do you want to set up the default disk group for each system? [y,n,q,?] (y) █
```

Figure 4-17 VERITAS Storage Foundation 4.0 default disk group

15. To set up the default disk group, type `y`. Press Enter to continue. You will see a window similar to Figure 4-18.

```
Specify a default disk group for system bc3srv4 or type 'l' to display a
listing of existing disk group(s). [?] pdc dg
```

Figure 4-18 VERITAS Storage Foundation 4.0 default group

16. Specify a default disk group. In this example, we typed `pdc dg`. Press Enter to continue. You will see a window similar to Figure 4-19.

```
You have decided to specify the default disk group as follows:

Host: bc3srv4 ..... Disk group: pdc dg

Note: If 'nodg' is displayed, then the host will be configured to have no
default disk group.

Is this correct? [y,n,q] (y)
```

Figure 4-19 VERITAS Storage Foundation 4.0 default disk group

17. Type `y` to indicate that the information in Figure 4-19 is correct. Press Enter to continue. You will see a window similar to Figure 4-20.

```
Setting default disk group to pdc dg on bc3srv4 ..... Done

Preparing to start daemons on target systems...
Starting vxrelocd on bc3srv4 ..... Success
Starting vxcached on bc3srv4 ..... Success
Starting vxconfigbackupd on bc3srv4 ..... Success

Storage Foundation Enterprise was started successfully.

Press [Enter] to continue:
```

Figure 4-20 VERITAS Storage Foundation 4.0 daemons start

18. The daemons are started. Press Enter. You will see a window similar to Figure 4-21 on page 42.

```
root@bc3srv4:/mnt/cdrom/rhel3_i686
Installation of Storage Foundation Enterprise 4.0 has completed
successfully.

The installation summary is saved at:

    /opt/VRTS/install/logs/installer817151755.summary

The installer log is saved at:

    /opt/VRTS/install/logs/installer817151755.log

The installation response file is saved at:

    /opt/VRTS/install/logs/installer817151755.response

CPI WARNING V-9-112-1030 While installing VxVM, the rpm(8) command generated
either an error or warning message. The message for each relevant system is
logged in the summary file specified above.

See the VERITAS File System Administrators Guide for information on using
VxFS.
[root@bc3srv4 rhel3_i686]#
```

Figure 4-21 VERITAS Storage Foundation 4.0 installation complete

The installation of VERITAS Storage Foundation 4.0 is now complete.

### 4.2.3 VERITAS Storage Foundation 4.0 post installation

After installing VERITAS Storage Foundation 4.0, the VERITAS Volume Manager requires the installation of the Array Support Library so that VERITAS Dynamic Multi Pathing can understand the array features.

In this example, we downloaded the Array Support Library from the VERITAS support site:

<http://support.veritas.com>

**Note:** The Array Support Library is documented in VERITAS technote 260741 (also found on the VERITAS support site). If there are any questions related to the Array Support Library, contact VERITAS support.

After you have downloaded the Array Support Library to your Linux server, execute the commands shown in Figure 4-22 on page 43.



```

Command Prompt - telnet linux
Linux # tar xvzf IBM_FAStT_260741.tar.Z
VRTSFAStT-1.0-1.0.i686.rpm
Linux # rpm -ihv VRTSFAStT-1.0-1.0.i686.rpm

```

Figure 4-22 Installing Array Support Library

The Array Support Library is provided for specified storage subsystems such as the FAStT. The storage subsystem should be certified by VERITAS. Certified storage subsystems are listed in VERITAS Hardware Compatibility List found at the following Web address:

<http://support.veritas.com/docs/271281>

The Hardware Compatibility List is a living document. It is updated on a regular basis. If your storage is not listed, contact your VERITAS sales person or VERITAS support for the latest information.

After installing the Array Support Library, reboot the server to make sure that everything works. To determine that Dynamic Multi Pathing is enabled, execute the `vxddmpadm` command as shown in Figure 4-23.

```

Command Prompt - telnet linux
Linux # vxddmpadm listctlr all
CTLR-NAME      ENCLR-TYPE      STATE      ENCLR-NAME
-----
c0             IBM_FAStT      ENABLED    IBM_FAStT0
c1             IBM_FAStT      ENABLED    IBM_FAStT0
c259          OTHER_DISKS     ENABLED    OTHER_DISKS
Linux # vxddmpadm getdmpnode enclosure=IBM_FAStT0
NAME           STATE           ENCLR-TYPE  PATHS  ENBL  DSBL  ENCLR-NAME
-----
sda            ENABLED        IBM_FAStT   2      2     0     IBM_FAStT0
sdb            ENABLED        IBM_FAStT   2      2     0     IBM_FAStT0
Linux #

```

Figure 4-23 Verifying Dynamic Multi Pathing

## 4.2.4 VERITAS Enterprise Administrator

VERITAS Enterprise Administrator (VEA) is provided for each Linux and Windows installation. In this example, we use VEA for Windows. To install VEA, follow these steps:

1. Insert the VERITAS Storage Foundation 4.0 CD-ROM into the Windows PC.
2. Open `drive_letter:\rhel3_i686\windows` using Windows Explorer. In this example, the drive letter is E.

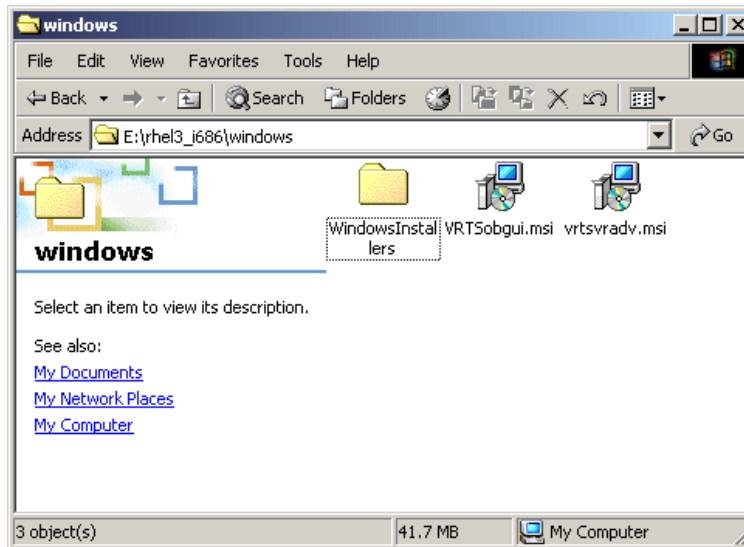


Figure 4-24 Place of VEA installer

3. Click **VRTSobgui.msi** to start the installer. Once the installer is running, click **Next**. The window shown in Figure 4-25 on page 45 will appear.

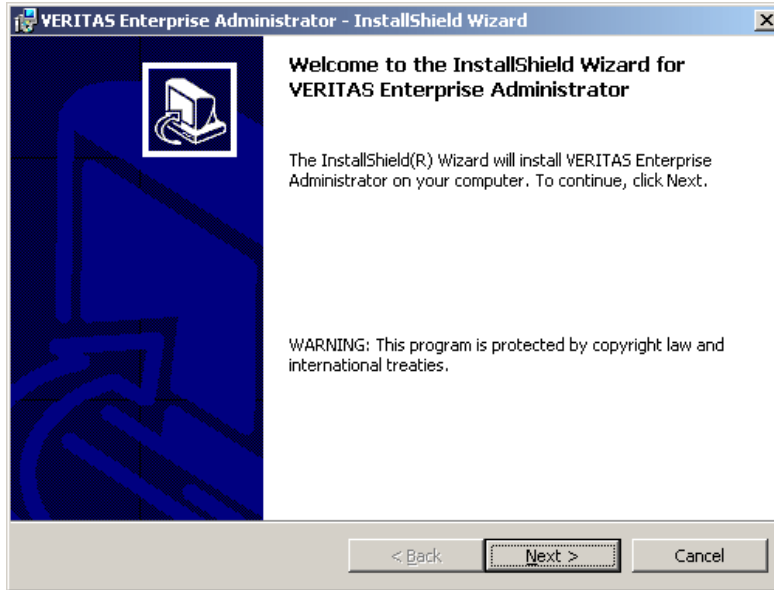


Figure 4-25 InstallShield Wizard

4. Click **Next**. The window shown in Figure 4-26 will appear.

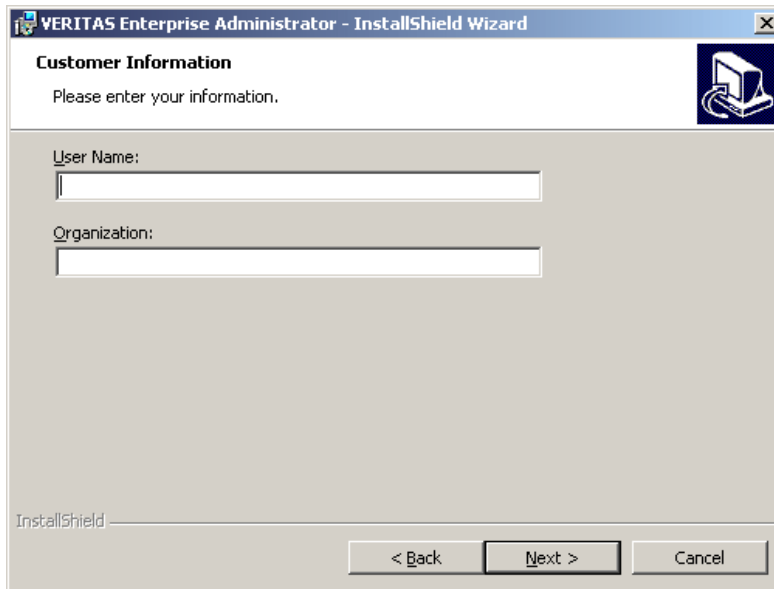


Figure 4-26 Customer Information window

5. Enter the User Name and Organization information. (It may be filled in automatically.) Click **Next**. The window shown in Figure 4-27 will appear.

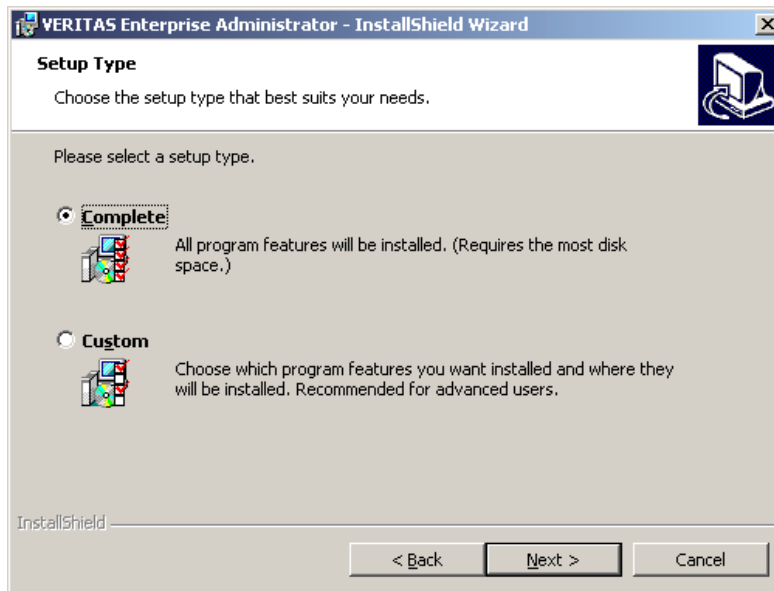


Figure 4-27 Setup Type window

6. Select **Complete** and click **Next**. The window shown in Figure 4-28 will appear.

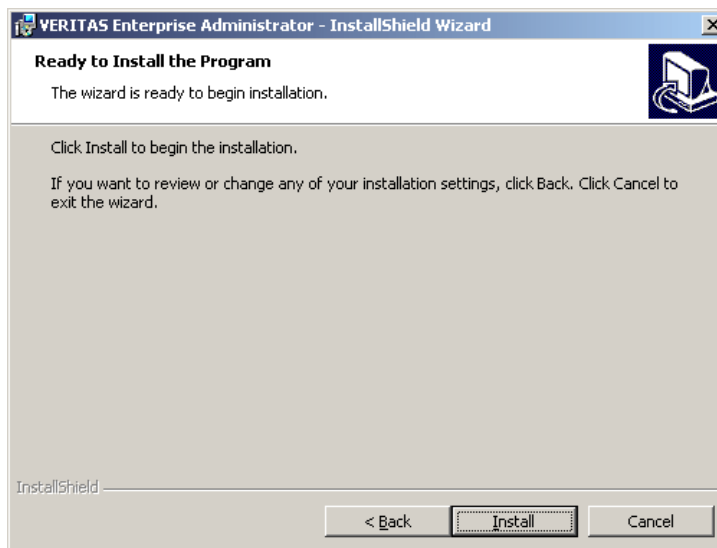


Figure 4-28 Ready to Install window

7. VEA installation is ready. Click **Install**. You will see the installation progress window as shown in Figure 4-29.

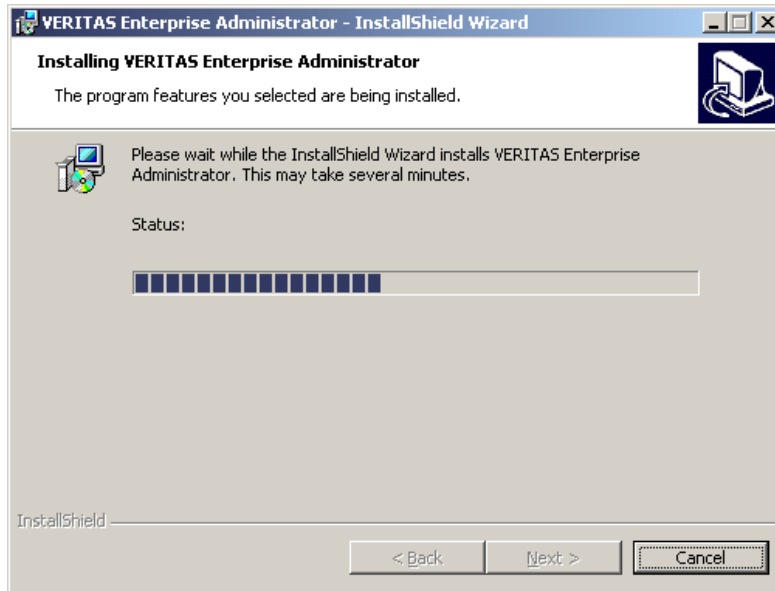


Figure 4-29 Installation progress window

8. When installation is complete, you will see the window in Figure 4-30.

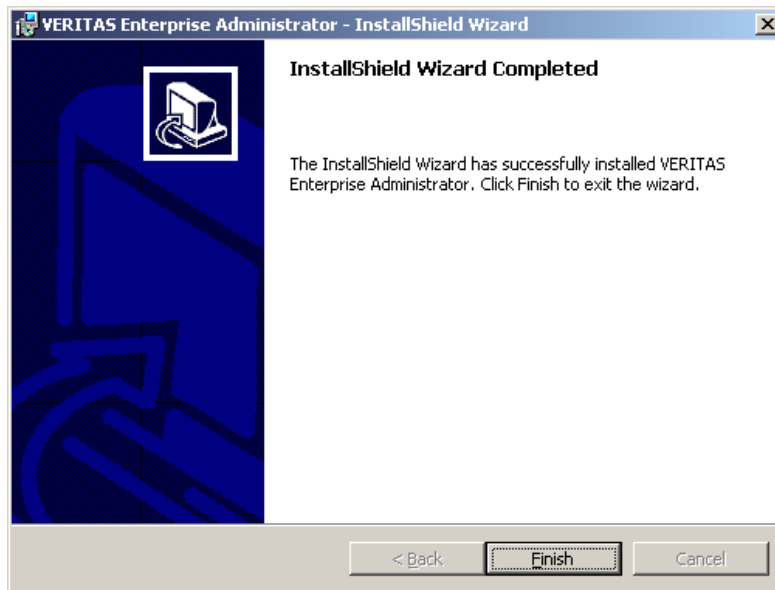


Figure 4-30 Installation complete window

9. Click **Finish**.

You start VERITAS Enterprise Administrator by clicking the desktop icon or selecting **Programs** → **VERITAS** → **VERITAS Enterprise Administrator** from the Windows Start menu.

After you start VERITAS Enterprise Administrator, you will see a window similar to Figure 4-31.

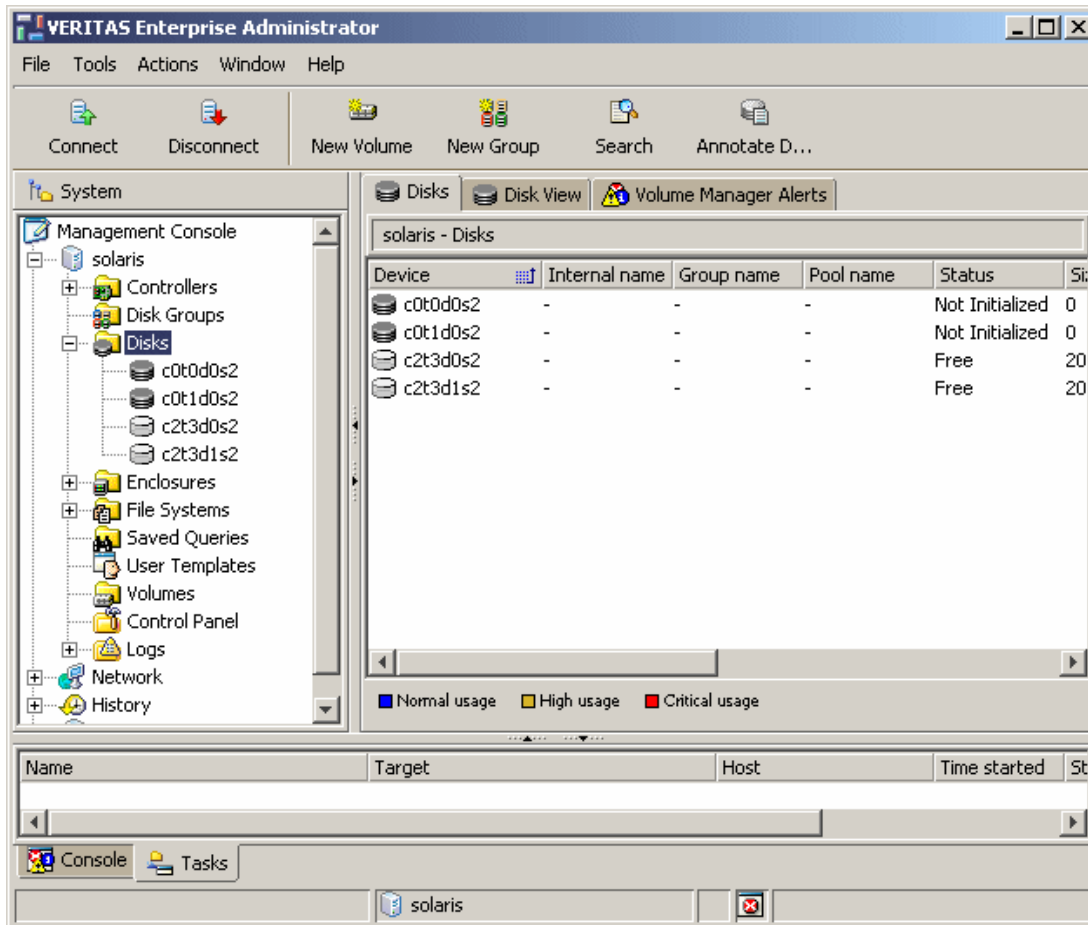


Figure 4-31 Volume Manager configuration before initializing disks

## 4.2.5 Initializing disks for a new file system

This section describes how to initialize disks and create a new file system using VEA. For more details on using VEA, see the *VERITAS Volume Manager 4.0 User's Guide - VERITAS Enterprise Administrator* that comes with the product software.

### Creating a new disk group

You should create a new disk group before creating the new volume. In this example, we used the following New Disk Group Wizard window to create a new disk group.

1. From the VERITAS Enterprise Administrator window (Figure 4-31 on page 48), click the **New Group** button. The window shown in Figure 4-32 will appear.

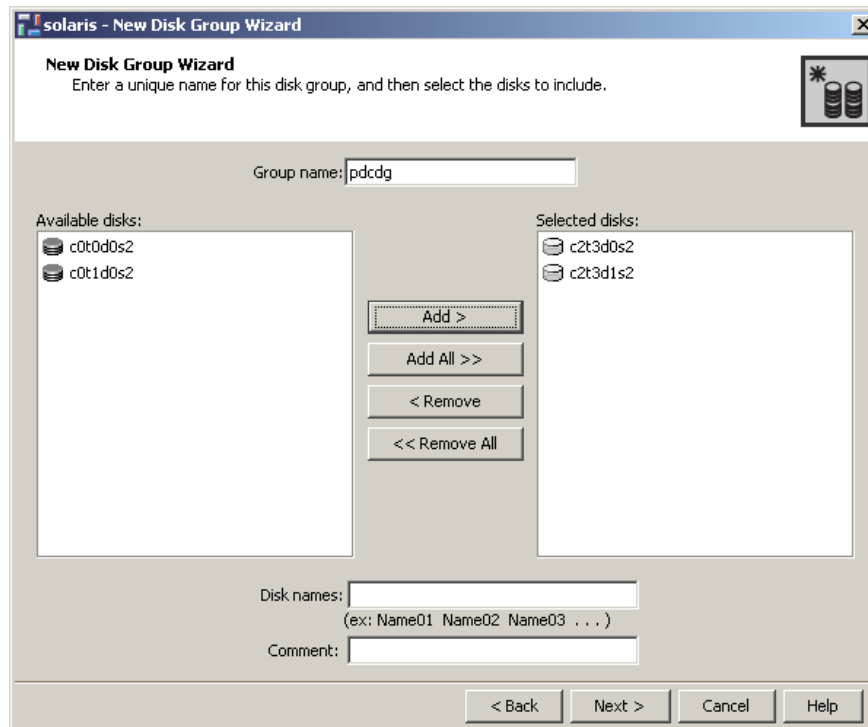


Figure 4-32 Disk selection for creating new diskgroup

2. In the New Disk Group Wizard window, you select disks on FASTT600 to include in the new disk group called *pdcdg*. Select the disks, and click **Add**. Click **Next**. The Organization principle window appears, as shown in Figure 4-33.

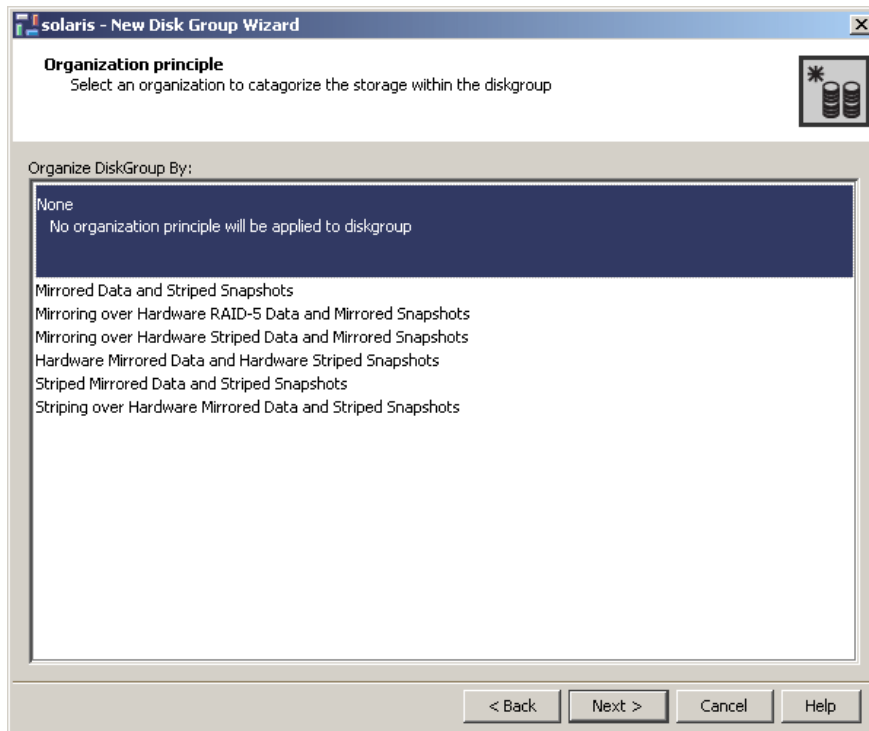


Figure 4-33 Organization principle selection for creating new diskgroup

3. In this window, in almost all cases, it is better to choose **None**. Select **None**, and click **Next**. A summary of the new disk group is shown (Figure 4-34 on page 51).



VERITAS Volume Manager has an Intelligent Storage Provisioning (ISP) feature. ISP is needed to create an ISP-style volume. Other items should be selected to create it. For more details about ISP, see the *VERITAS Volume Manager 4.0 User's Guide* that comes with the product software.

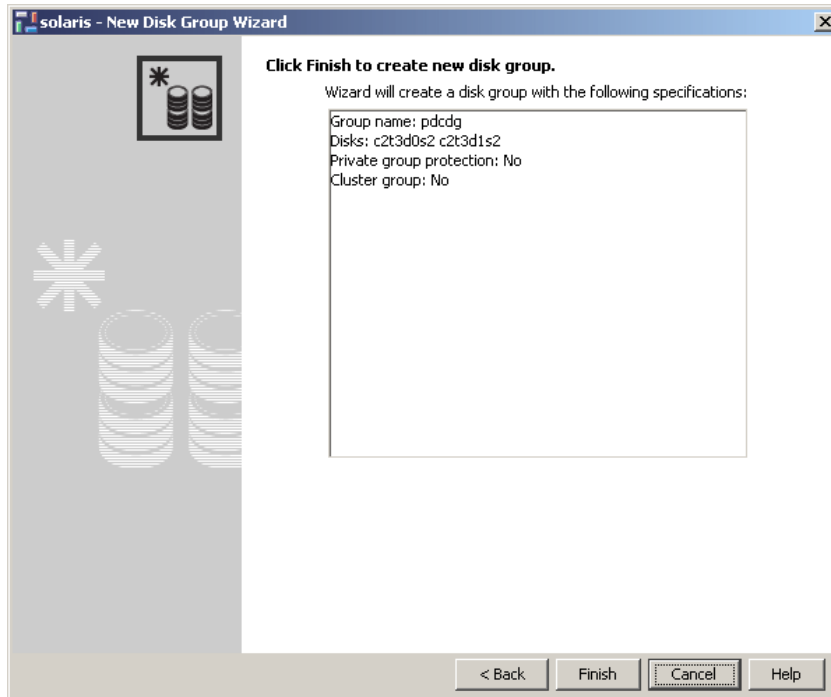


Figure 4-34 Selection summary for creating new diskgroup

- Click **Finish** to create new disk group. A window similar to that shown in Figure 4-35 will appear. The Internal name of each disk is created and named automatically.

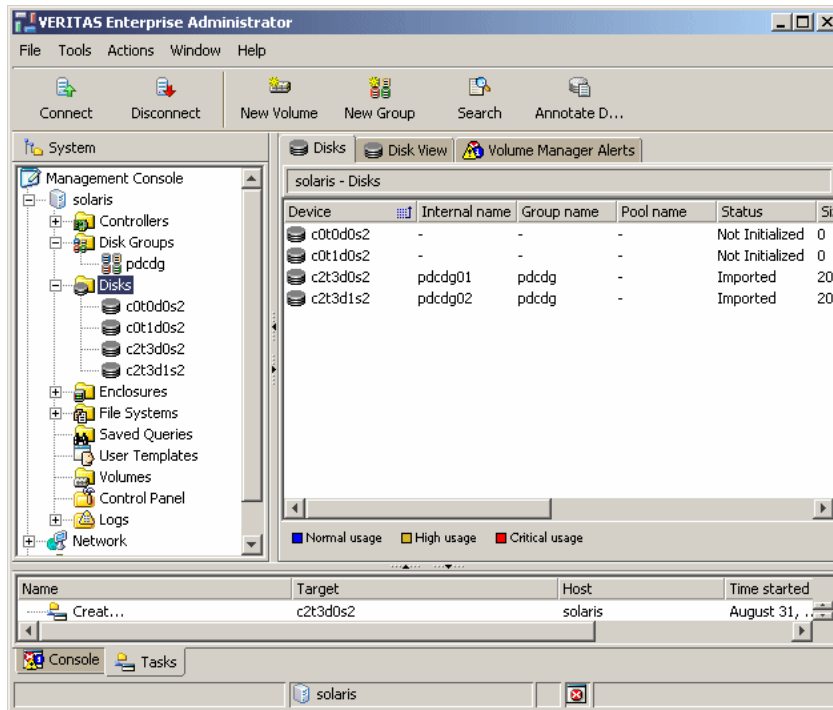


Figure 4-35 Created new diskgroup

## Creating a new volume

From the VERITAS Enterprise Administrator window (Figure 4-31 on page 48), click **New Volume** to create a new volume. You will see a window similar to Figure 4-36 on page 53.

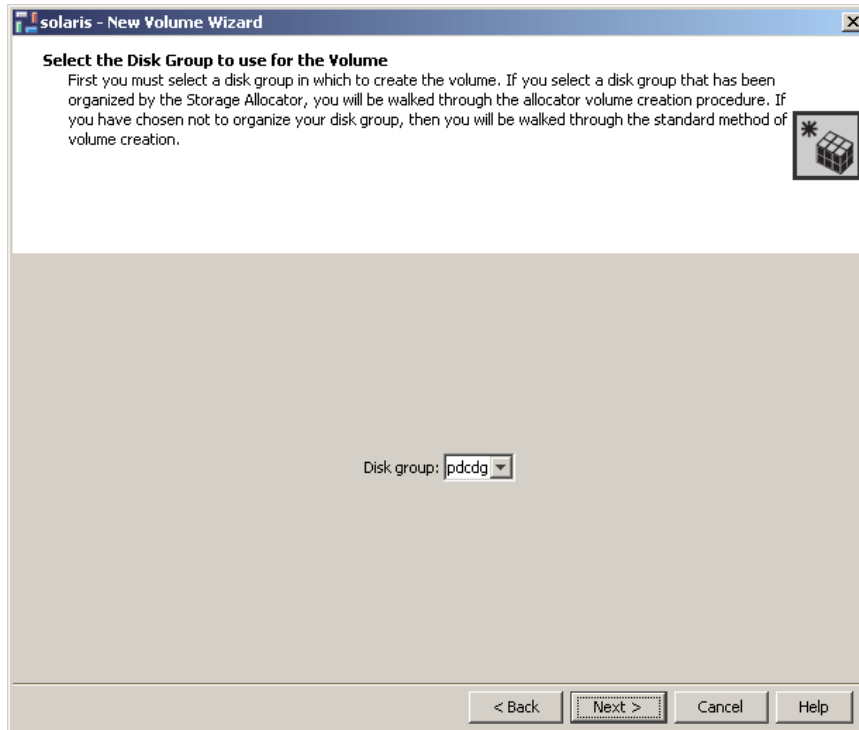


Figure 4-36 Diskgroup selection for creating new volume

1. This window shows an existing disk group in which the volume is created. In this example, we selected **pdc dg** to create new volume. Click **Next** to continue.

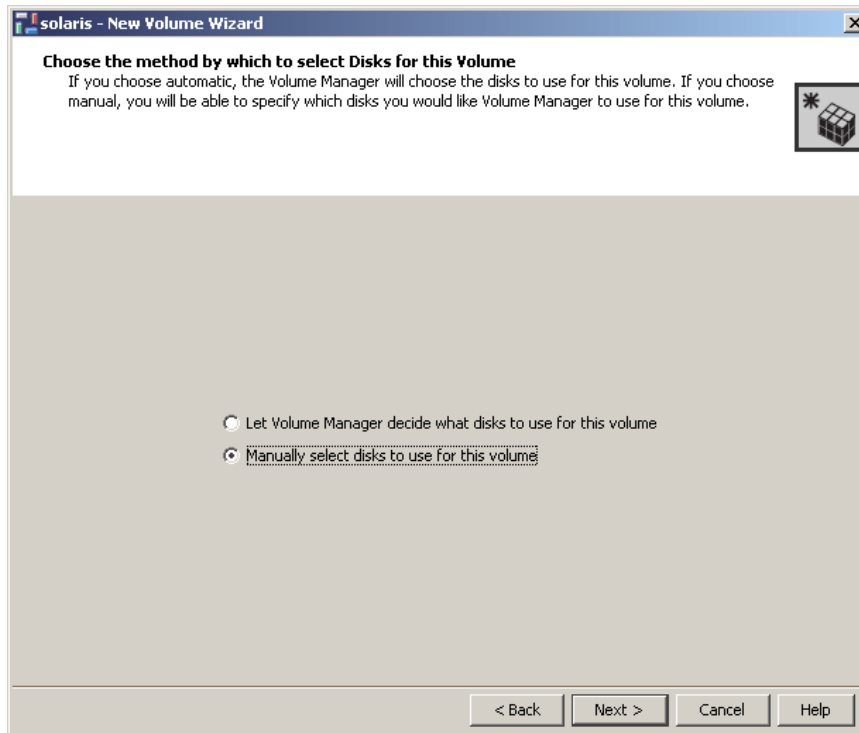


Figure 4-37 Disk selection for creating new volume

2. The manual disk selection is needed to create the volume. Choose **Manually select disk to use for this volume**. Click **Next**. You will see a window similar to Figure 4-38 on page 55.

**Note:** If you choose **Let Volume Manager decide what disks to use for this volume** in Figure 4-37, you would skip this step.

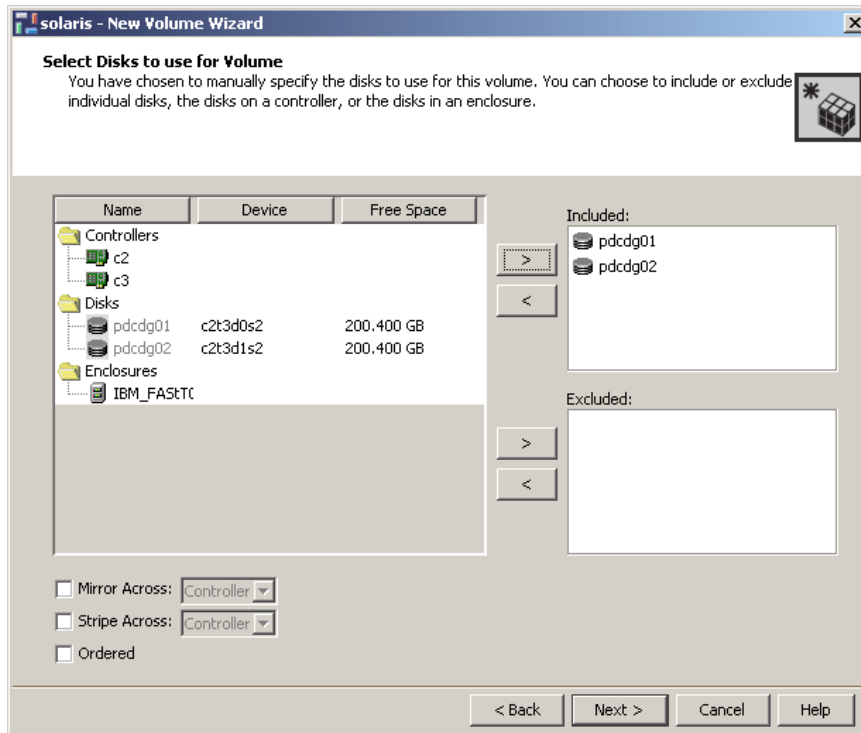


Figure 4-38 Disk selection for creating new volume continue

3. Select the disk for your volume and click **Next**. For this example, we selected **pdcdg01** and **pdcdg02**.

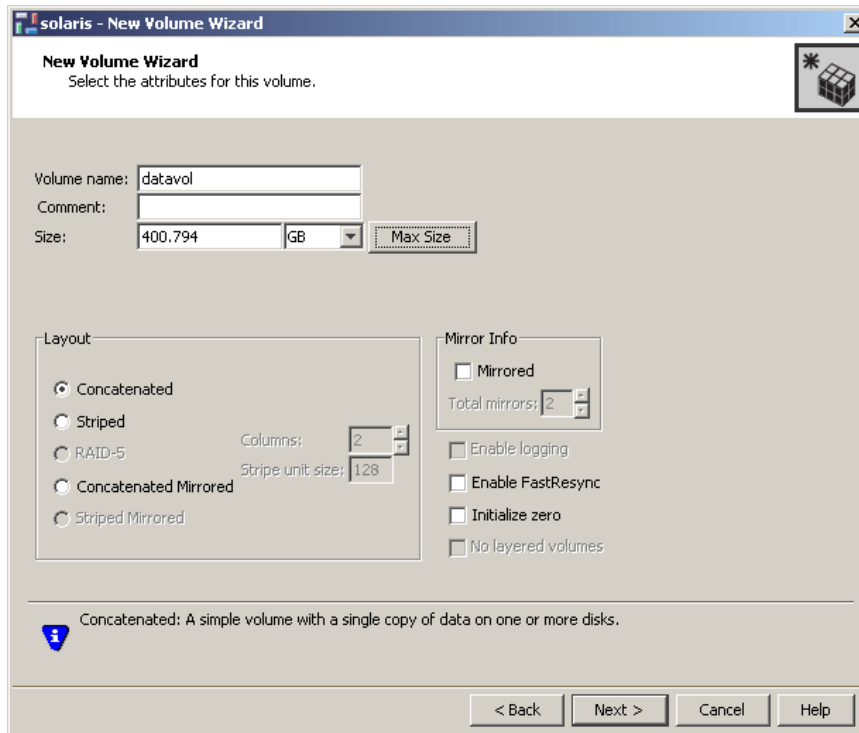


Figure 4-39 Name and size definition for creating new volume

4. You define the name and the size of the new volume in Figure 4-39. In this example, we entered `datavol` as our Volume name, and we used all the space available. Under the Layout area, select **Concatenated**. Click **Next**. You will see a window similar to Figure 4-40 on page 57.

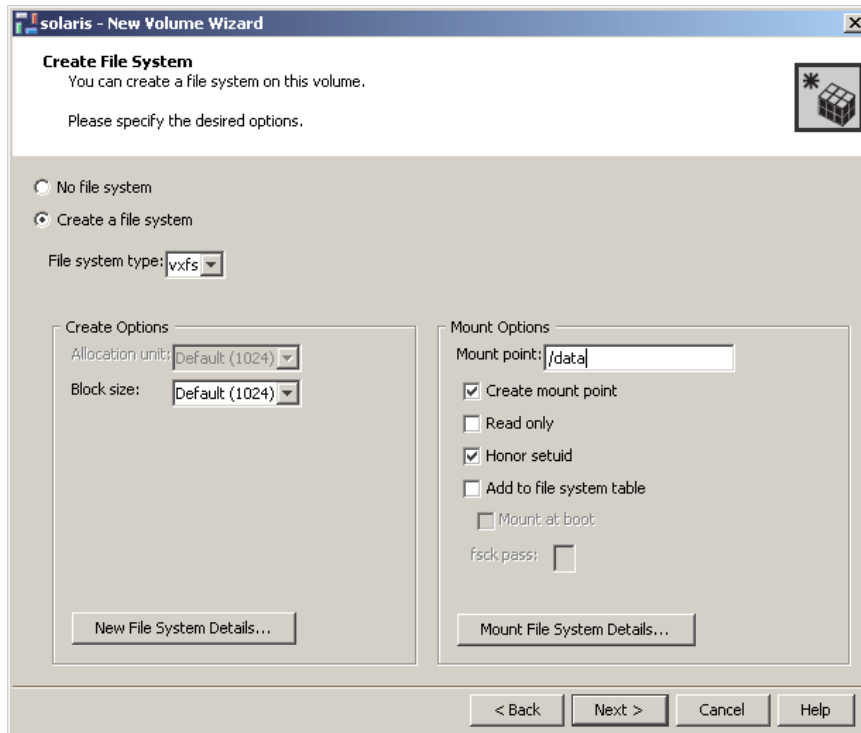


Figure 4-40 File system definition for creating new volume

5. In Figure 4-40, you should define the file system type and mount point for creating the file system on the volume. Select **Create a file system**. Then select **vxfs** as the file system type. Under the Mount Options Mount point type /data. Click **Create mount point** and **Honor setuid**. Click **Next** to continue. A window similar to Figure 4-41 on page 58 will appear showing a summary.

**Note:** In this example, we selected the vxfs file system type to allow for increasing and decreasing file system size without unmounting the file system.

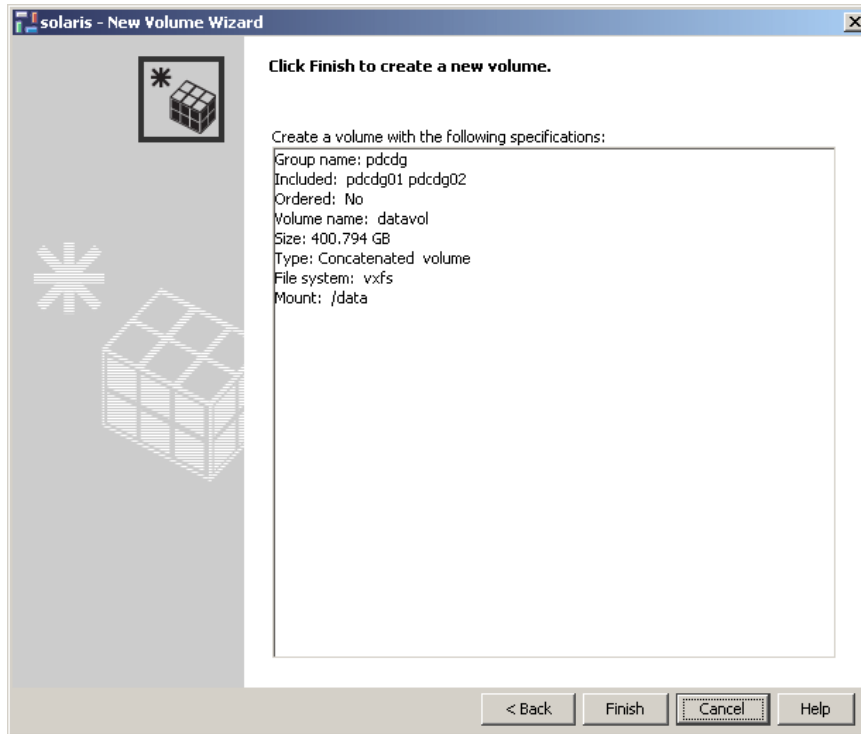


Figure 4-41 Selection summary for creating new volume

6. Click **Finish** to create new disk group.

Figure 4-42 on page 59 shows that the volume `datavol` was created and mounted to a specified mount point.



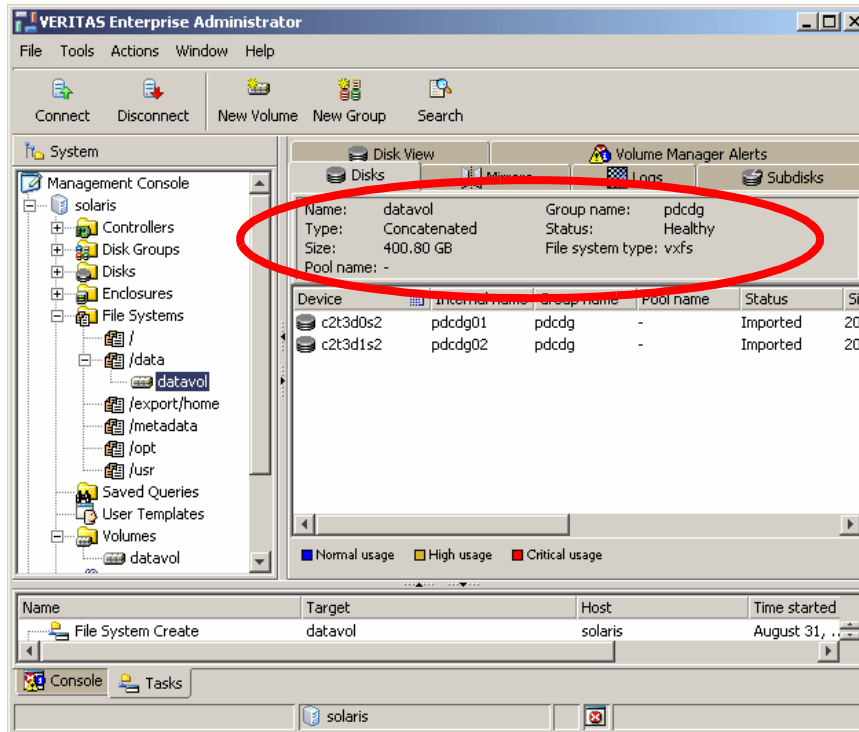


Figure 4-42 Created new volume

## Creating data for the migration example

For the example, 100 000 files and a plain text file were created in the directory /data. The file size, except for the plain text file, is 10 KB, and 1 000 files were placed in a directory. At first, we added the following phrase in the plain text file using an editor:

Veritas liberabit vos:  
 This phrase means “The truth will set you free“ in English.

Next, we created 100 000 files using the script in Example 4-1.

### Example 4-1 The script to create files

```
#!/bin/sh

TARGET=/data
DIR_BASENAME=s_dir
FILE_BASENAME=s_file

# Total size = SIZE_BASE x DD_COUNT x FILES_PER_DIR x DIR_LIMIT
```

```

FILES_PER_DIR=1000
DIR_COUNT=1
DIR_LIMIT=100
SIZE_BASE=1000
DD_COUNT=10

create_file ()
{
    FILE_COUNT=`expr \( $1 - 1 \| \) \* $FILES_PER_DIR + 1`
    FILE_LIMIT=`expr $1 \| \* $FILES_PER_DIR`

    while [ $FILE_COUNT -le $FILE_LIMIT ];
    do
        FILE=$2/$FILE_BASENAME$FILE_COUNT
        dd if=/dev/zero of=$FILE bs=$SIZE_BASE count=$DD_COUNT > /dev/null 2>&1
        FILE_COUNT=`expr $FILE_COUNT + 1`
    done
}

while [ $DIR_COUNT -le $DIR_LIMIT ];
do
    DIR_NAME=$TARGET/$DIR_BASENAME$DIR_COUNT
    mkdir $DIR_NAME
    create_file $DIR_COUNT $DIR_NAME &
    DIR_COUNT=`expr $DIR_COUNT + 1`
done
wait

```

---

**Note:** We used this script only to create the data for our migration. You should not use this script for other purposes.

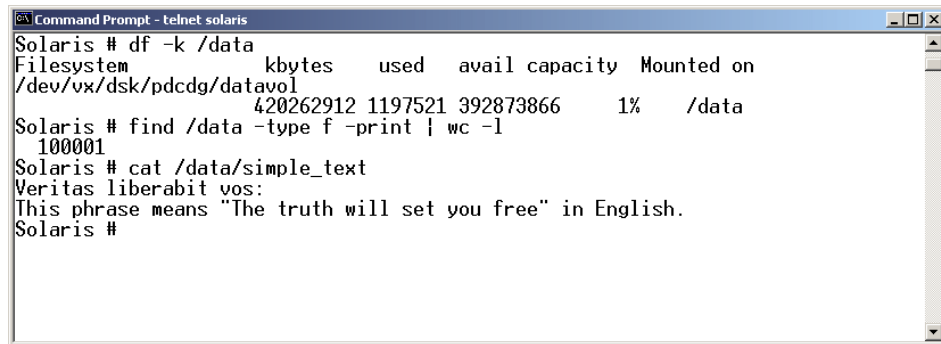
## 4.2.6 Starting with an existing Volume Manager or File System setup

If the system already has Volume Manager and File System installed and if the version is earlier than 4.0, you must follow the upgrade procedure. This procedure is documented in *VERITAS Storage Foundation 4.0 Installation Guide* that comes with the product software.

If the version is 4.0 or later, and if the File System version as well as the Volume Manager disk layout is upgraded to the latest version, you can proceed to 4.2.7, “Step-by-step migration” on page 61.

## 4.2.7 Step-by-step migration

Figure 4-43 verifies the example data before migration. For this example, we created 100 000 files and a plain text file in the /data directory.



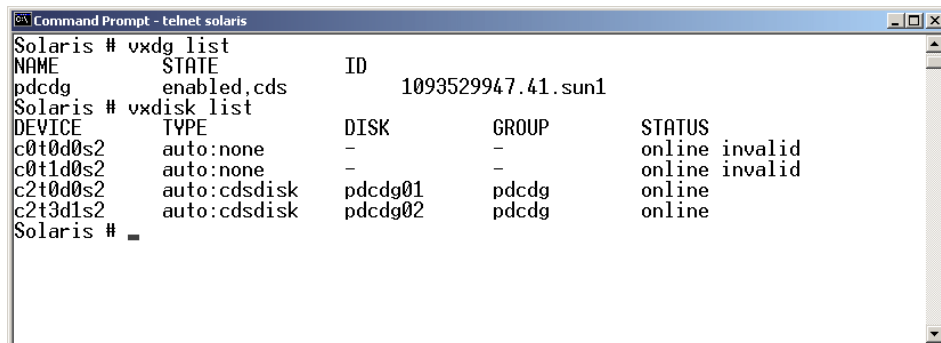
```
Command Prompt - telnet solaris
Solaris # df -k /data
Filesystem              kbytes    used    avail capacity  Mounted on
/dev/vx/dsk/pdcdg/datavol
                        420262912 1197521 392873866    1%    /data
Solaris # find /data -type f -print | wc -l
100001
Solaris # cat /data/simple_text
Veritas liberabit vos:
This phrase means "The truth will set you free" in English.
Solaris #
```

Figure 4-43 Verifying data on Solaris server

The commands to run the verification are:

```
df -k /data
find /data -type f -print | wc -l
cat /data/simple_text
```

Figure 4-44 verifies the diskgroup and disk on Solaris. For example, the diskgroup is created as pdcdg. The diskgroup has two disks, pdcdg01 and pdcdg02. Disk pdcdg01 corresponds to the logical drive 1, and disk pdcdg02 corresponds to logical drive 2 on FAST600.



```
Command Prompt - telnet solaris
Solaris # vxvg list
NAME      STATE          ID
pdcdg     enabled,cds    1093529947.41.sun1
Solaris # vxdisk list
DEVICE    TYPE          DISK          GROUP         STATUS
c0t0d0s2  auto:none     -             -             online invalid
c0t1d0s2  auto:none     -             -             online invalid
c2t0d0s2  auto:cdsdisk pdcdg01       pdcdg         online
c2t3d1s2  auto:cdsdisk pdcdg02       pdcdg         online
Solaris #
```

Figure 4-44 Verifying diskgroup and disk on Solaris server

The commands to run the verification are:

```
vxvg list
vxdisk list
```

Figure 4-45 verifies the volume on Solaris. For example, the volume is created as `datavol`, and the volume uses all the disks in the diskgroup which it belongs. The volume is made by concatenated disks, not mirrored disks.

```

Command Prompt - telnet solaris
Solaris # vxprint
Disk group: pdc dg

TY NAME      ASSOC      KSTATE  LENGTH  PLOFFS  STATE  TUTILO  PUTILO
dg pdc dg    pdc dg    -        -        -        -        -
dm pdc dg01  c2t3d0s2  -        420263680 -        -        -
dm pdc dg02  c2t3d1s2  -        420263680 -        -        -
v datavol    fsgen     ENABLED  840525824 -        ACTIVE  -        -
v datavol-01 datavol   ENABLED  840525824 -        ACTIVE  -        -
sd pdc dg01-01 datavol-01 ENABLED  420263680 0        -        -
sd pdc dg02-01 datavol-01 ENABLED  420262144 420263680 -        -
Solaris #

```

Figure 4-45 Verifying volume on Solaris server

The command to run the verification is:

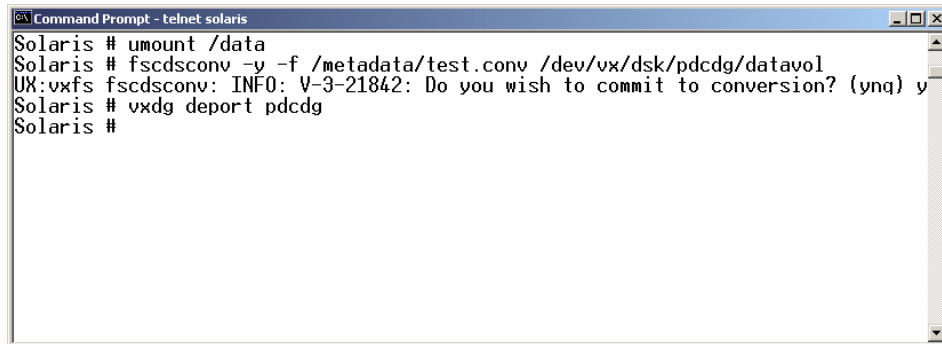
```
vxprint
```

The next is the most important step of the migration using Portable Data Containers. You should unmount the target file system before the file system conversion. After unmounting, the file system is converted using the `fscdsconv` command. The information of the file system is saved as a recovery file (metadata).

The recovery file size is difficult to predict. The size of the recovery file depends on the number of files and directories which the file system includes. See 6.3, “Calculating the space requirements of the conversion process” on page 81 for best practices regarding recovery file space requirements.

After the conversion, you should deport the diskgroup pcdcg using the **vx dg deport** command for migration. To deport the diskgroup, follow these steps:

1. From a Solaris command prompt, run **umount /data** (see Figure 4-46).



```
Command Prompt - telnet solaris
Solaris # umount /data
Solaris # fscdsconv -y -f /metadata/test.conv /dev/vx/dsk/pcdcg/datavol
UX:vxfs fscdsconv: INFO: V-3-21842: Do you wish to commit to conversion? (ynq) y
Solaris # vx dg deport pcdcg
Solaris #
```

Figure 4-46 File system conversion and deport diskgroup

2. Next, run **vx dg deport pcdcg**.

After deporting the diskgroup from Solaris, you should change the logical drive mapping to import the migrated diskgroup to Linux. Linux requires a reboot after changing logical drive mapping (Figure 4-47).

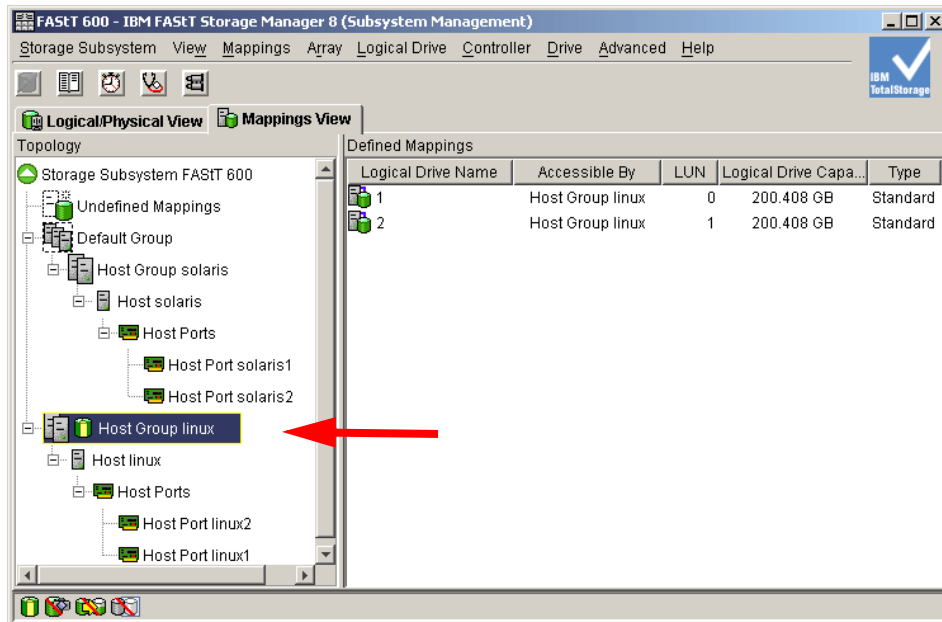
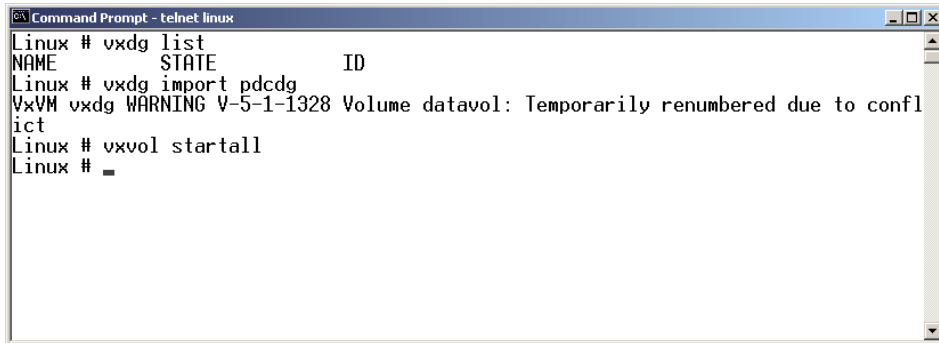


Figure 4-47 Changed logical drive mapping

After rebooting the Linux server, import the diskgroup to the Linux server. For example, the diskgroup pdc dg which was migrated from the Solaris server is imported in Figure 4-48.



```
Command Prompt - telnet linux
Linux # vxdg list
NAME          STATE          ID
Linux # vxdg import pdc dg
VxVM vxdg WARNING V-5-1-1328 Volume datavol: Temporarily renumbered due to conflict
Linux # vxvol startall
Linux # _
```

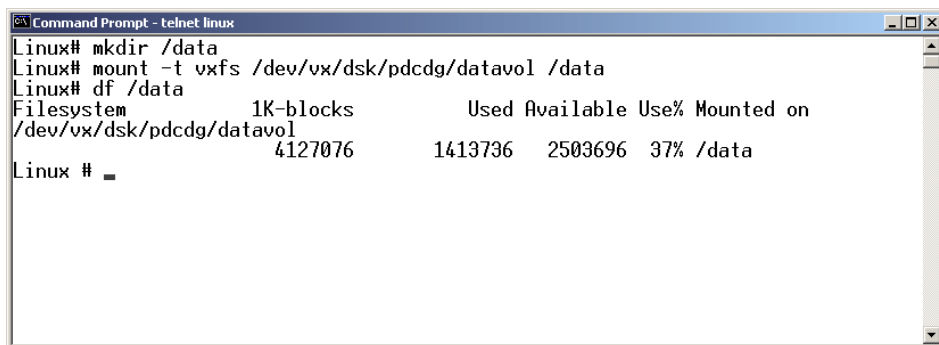
Figure 4-48 Import diskgroup and start volume on Linux

3. Verify the existence of the diskgroup, and then prepare to import by running the following commands:

```
vx dg list          (group should not exist)
vx dg import pdc dg (imports the diskgroup)
vx vol starall     (starts all volumes)
```

**Note:** The warning message does not matter to importing the diskgroup. You can ignore it.

4. After you have imported the diskgroup and have started the volume, you can mount the file system on the volume on an appropriate mount point (Figure 4-49).



```
Command Prompt - telnet linux
Linux# mkdir /data
Linux# mount -t vxfs /dev/vx/dsk/pdc dg/datavol /data
Linux# df /data
Filesystem          1K-blocks          Used Available Use% Mounted on
/dev/vx/dsk/pdc dg/datavol
                    4127076           1413736   2503696   37% /data
Linux # _
```

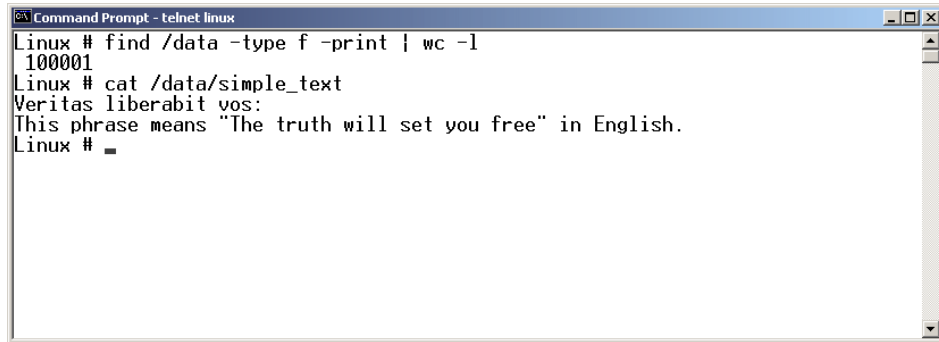
Figure 4-49 Mount file system on Linux

The commands executed are:

```
mkdir /data  
df /data
```

5. Finally, the data has been migrated. In Figure 4-50, verify the migration of the data by running the following commands:

```
find /data -type f -print | wc -l  
cat /data/simple_text
```

A screenshot of a telnet session window titled "Command Prompt - telnet linux". The window shows a Linux shell prompt. The user enters the command "find /data -type f -print | wc -l", and the output is "100001". The user then enters "cat /data/simple\_text", and the output is "Veritas liberabit vos:  
This phrase means 'The truth will set you free' in English." The prompt returns to "Linux # \_".

```
Linux # find /data -type f -print | wc -l  
100001  
Linux # cat /data/simple_text  
Veritas liberabit vos:  
This phrase means "The truth will set you free" in English.  
Linux # _
```

Figure 4-50 Verifying data on Linux







## Performance

This chapter discusses the performance aspect of using VERITAS Portable Data Containers. It shows that Portable Data Containers scale with the number of files and the amount of data. This chapter also compares the throughput of Portable Data Containers with theoretical network throughput and disk transfer speed.

## 5.1 Data migration performance

The time it takes to migrate data is generally one of the biggest challenges for a successful migration. Because most computing environments today are running 24x7, shutting down servers for a few hours or even days to migrate data can be very costly and sometimes even impossible.

VERITAS Portable Data Containers can eliminate this challenge. As mentioned earlier, Portable Data Containers do not touch the data in the file system. Thus, the time it takes to migrate data is contingent upon the amount of metadata in the file system.

The amount of metadata in a file system primarily depends on three factors:

- ▶ The size of the file system
- ▶ The amount of files
- ▶ Fragmentation of file system

The administrator cannot modify the first two factors in this list. The administrator can, however, modify the third. VERITAS File System has the capability to do online defragmentation. Depending on the environment, VERITAS recommends that you defragment nightly or weekly. You can configure VERITAS File System to run a specific amount of time so it does not affect daily operations. You can also optimize VERITAS File System to run only with various parameters. For more details regarding defragmenting the file system, see Appendix A, “Manual Pages” on page 123.

A large file may generate a small amount of additional metadata. (The file system may be required to keep track of additional extents. It depends on how the file was created and allocated.) However, in our testing for this redbook, we noticed no difference when migrating small files versus large files.

With Portable Data Containers, it is important to remember how long it can take to migrate a file system. A definite time to convert a file system cannot be given because every file system is different. However, we ran a number of tests so that you can compare your setup with the test setup and draw useful conclusions.

We determined that we needed to test two types of environments to be able to give guidance for how long a conversion can take. We performed:

- ▶ Scalability tests to show how Portable Data Containers perform in a file system with a large amounts of files
- ▶ Throughput tests to show how Portable Data Containers perform in a environment with large files

We used the same lab environment as documented in Chapter 4, “VERITAS Portable Data Containers for migration” on page 25.

We generated the files in these tests using the script supplied in Appendix A, “Manual Pages” on page 123. Because all files were created in a new file system so that there was no fragmentation present, we had the most optimal environment possible.

### 5.1.1 Data collection

During the performance tests, we took the following measurements to better understand the behavior of Portable Data Containers and to be able to report on how much resources were consumed by the migration:

- ▶ **Time.** The time it takes to do the actual conversion, including the reading of the file system, creating the recovery file, and writing the new metadata into the file system. Time was measured with the `time` command.
- ▶ **CPU usage.** Measured with the `mpstat` and `iostat` utilities. Both of these utilities are provided with the Solaris operating environment. The resulting graph provides the total CPU usage in the system because we used a dual processor server.
- ▶ **I/O usage.** Measured with `vxstat`. `vxstat` is part of VERITAS Volume Manager and provides detailed I/O statistics down to the subdisk level.
- ▶ **Recovery file size.** We also collected the size of the recovery file. We used this data in Chapter 6, “Best practices” on page 79.

## 5.2 Scalability

It is important to understand that Portable Data Containers are a *passive* technology. They do not affect performance while the file system is mounted and the server is in production. They are used only when actually migrating data. This design means that the concept of scalability only applies when actually migrating data.

The numbers shown in this chapter have nothing to do with the scalability of the VERITAS File System or VERITAS Volume Manager, but are provided as a guidance for how long a migration may take with the number of files tested.

To measure the scalability of Portable Data Containers, we executed a number of tests as shown in Table 5-1. In each of these tests, the file size was 10 KB.

*Table 5-1 Scalability test, number of files and amount of data*

<b>Test</b>	<b>Number of files/directories</b>	<b>Amount of data</b>
Workgroup	100 000/100	1.1 GB
Large workgroup	500 000/500	5.3 GB
Enterprise	1 000 000/1000	10.5 GB
Medium enterprise	5 000 000/5000	51.6 GB
Large enterprise	10 000 000/10,000	103 GB

### **5.2.1 Scalability performance results**

Table 5-2 on page 71 and Figure 5-1 on page 71 show the results of the file scalability tests. What is noticeable is that the curve is not a linear curve until it reaches one million files, after which it is linearly scalable until ten million files (the maximum tested for this book). VERITAS has tested several hundred million files and has found no upper limit.

This part-linear scalability is due to the minimum amount of metadata structures present in a file system. As previously mentioned in this chapter, one of the major factors for the amount of metadata is the size of the file system. VERITAS File System requires a small percentage of the file system to be dedicated to a minimum amount of metadata structures. This means that in reality, the difference in converting a file system with 500 or 500 000 files is minimal, which is good news for the administrator.

Table 5-2 Scalability results, number of files and amount of data

Test	Number of files/directories	Conversion time
Workgroup	100 000/100	31 seconds
Large workgroup	500 000/500	46 seconds
Enterprise	1 000 000/1000	1 min, 38 sec
Medium enterprise	5 000 000/5000	7 min, 57 sec
Large enterprise	10 000 000/10 000	14 min, 50 sec

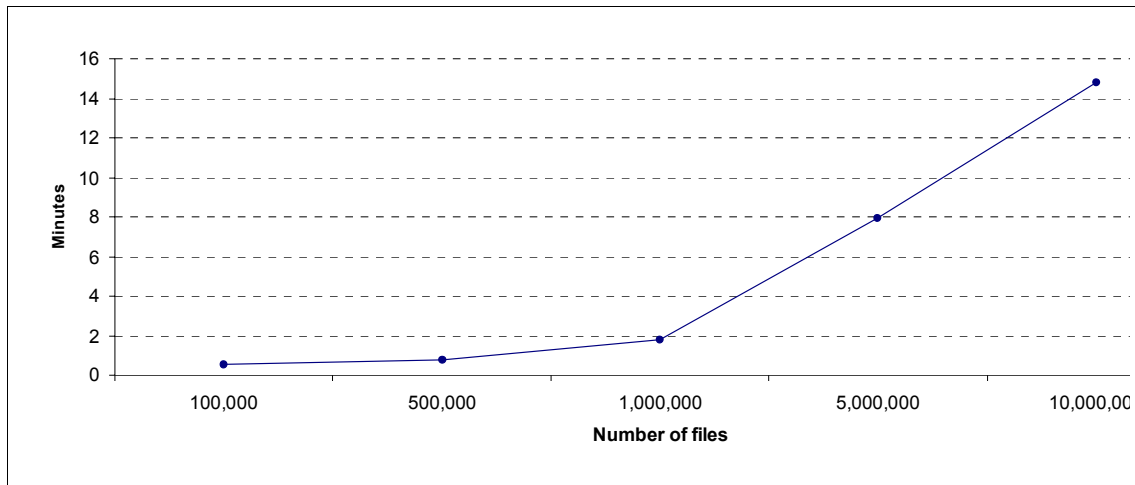


Figure 5-1 Portable Data Containers scalability graph

## 5.2.2 CPU and I/O

An important factor to consider is how much CPU and I/O bandwidth will be used during the actual migration. It is conceivable that only a single application and its data is migrated, leaving other applications on the original server to be migrated at a later time. Thus, it is important to understand the impact on CPU and I/O to make sure that the other applications are not affected.

Figure 5-2 on page 72 illustrates the effect on the CPU when using Portable Data Containers. Figure 5-3 on page 72 and Figure 5-4 on page 73 show the effect on the I/O throughput and usage.

In our lab environment, we used a fairly small server with two 450 MHz UltraSparc-II CPUs. In larger systems with more and faster CPUs, the migration process should be even less noticeable. However, it may not go faster. The

migration process can only read a certain amount of metadata at the same time, and the throughput to the recovery file can be a bottleneck as well. For information about the recovery file, see Chapter 6, “Best practices” on page 79.

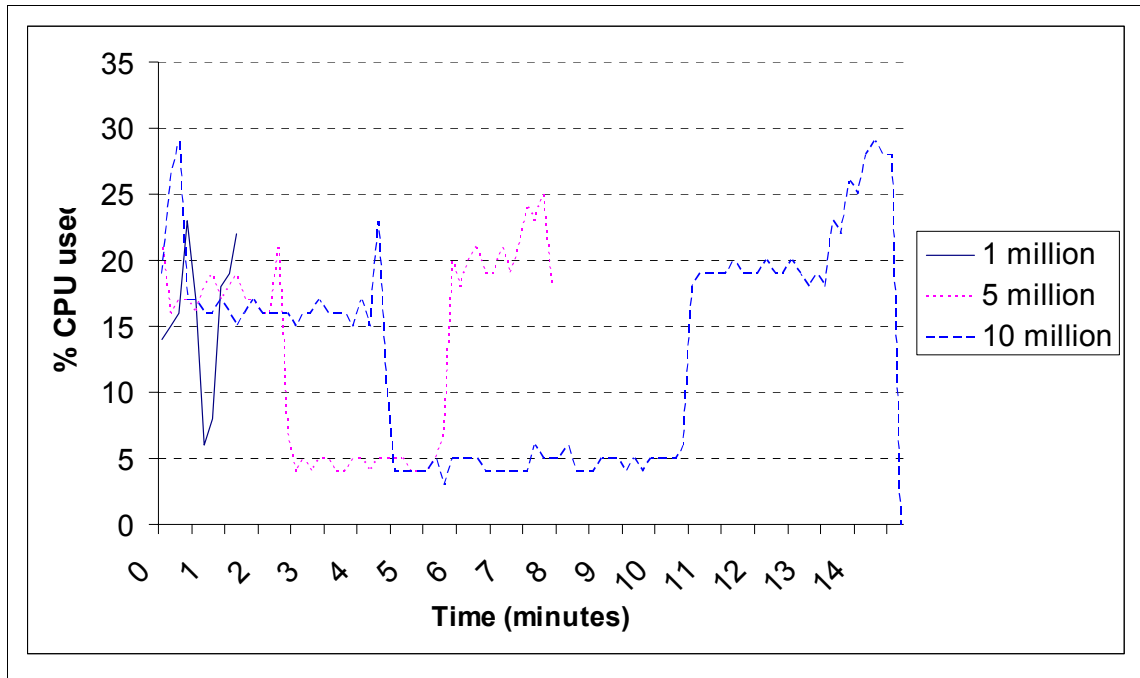


Figure 5-2 Portable Data Containers Scalability - CPU usage

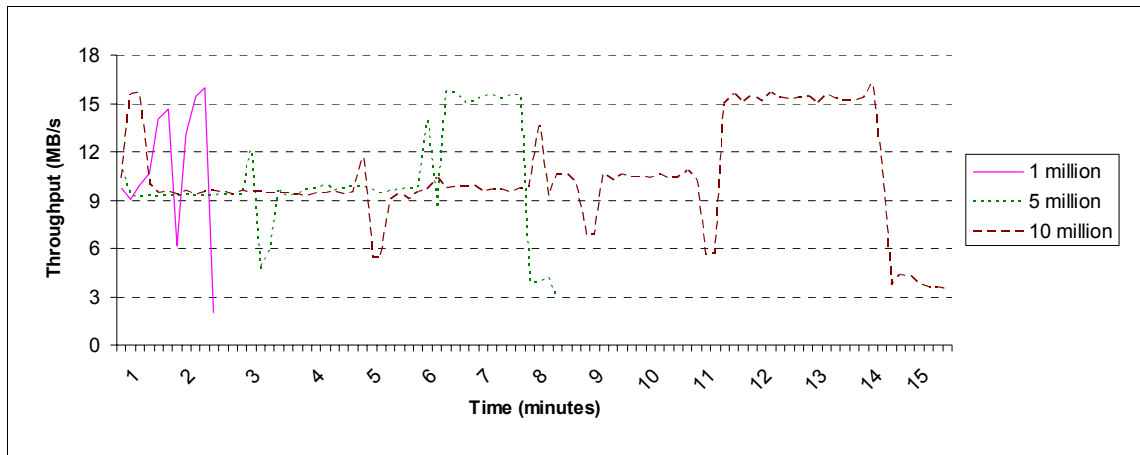


Figure 5-3 Portable Data Containers Scalability - I/O throughput

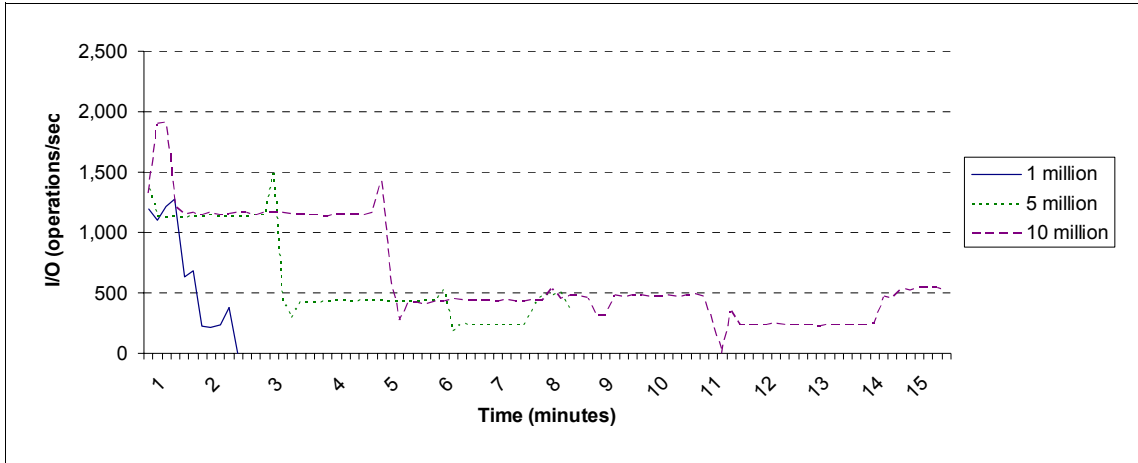


Figure 5-4 Portable Data Containers - I/O usage, operations per second

## 5.3 Throughput

The previous tests were focused on showing that Portable Data Containers can scale with the numbers of files (increased metadata amount). Scalability is only one part of performance measurements that are important while considering a new technology. I/O throughput is equally important.

As previously mentioned, Portable Data Containers do not actually convert the data but only convert the metadata. This means that the amount of user data (file size) is irrelevant to the conversion times for Portable Data Containers.

We executed the following five tests to illustrate the throughput of Portable Data Containers:

Test 2	10 MB of data, 1000 files
Test 3	100 MB of data, 1000 files
Test 4	1 GB of data, 1000 files
Test 5	10 GB of data, 1000 files
Test 6	100 GB of data, 1000 files

Keep in mind that throughput in this case is not an absolute value. If we had used a larger file system, the throughput would have been larger.

In these tests, we varied the file size to create a constant number of files while increasing the amount of data.

### 5.3.1 Throughput results

As shown in Table 5-3, a large amount of data can be converted very fast. What is interesting is that the difference between converting 10 MB of data and 100 GB of data is less than eight seconds. This further shows that the number of files affects the conversion more than the amount of data, which is very important for real customer migrations. (A graphical representation of this data is shown in Figure 5-5.)

As a side note, our conversion time for an empty 400 GB file system (in our test environment) took 18 seconds. This conversion was done without any files in the file system, illustrating that the minimum amount of metadata in a 400 GB file system will take 18 seconds to convert.

Table 5-3 Amount of data and number of files

Test	Amount of data	Number of files	Time to convert
Workgroup	10 MB	1000	19 seconds
Large workgroup	100 MB	1000	19.3 seconds
Enterprise	1 GB	1000	20.1 seconds
Medium enterprise	10 GB	1000	19.7 seconds
Large enterprise	100 GB	1000	26.6 seconds

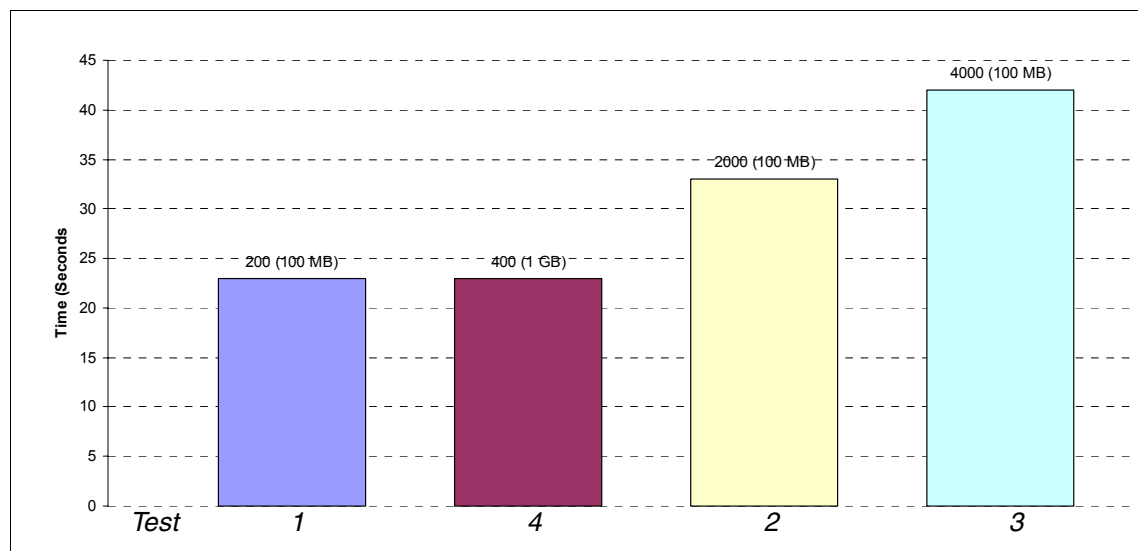


Figure 5-5 Portable Data Containers Throughput



### 5.3.2 CPU and I/O

Because the conversion is so fast, the I/O collection is not as extensive. However, as the numbers Figure 5-6 and Figure 5-7 on page 76 show, there is no real difference in CPU and I/O usage from converting a file system with millions of files.

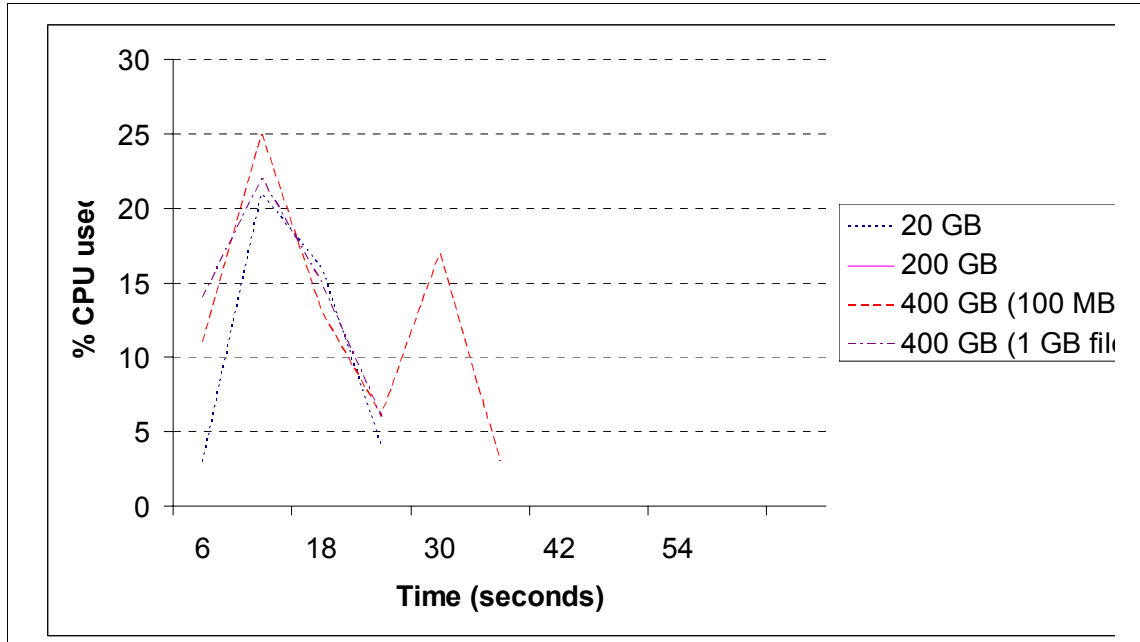


Figure 5-6 Portable Data Containers Throughput - CPU usage

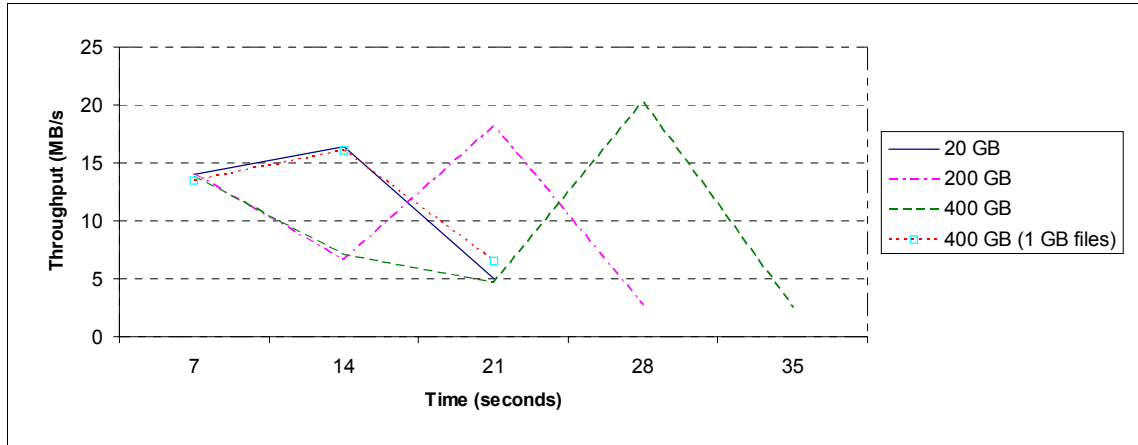


Figure 5-7 Portable Data Containers Throughput - I/O usage

## 5.4 Performance comparison

While Portable Data Containers is a great technology, it is also important to understand how it compares against other migration techniques. There are many different ways of moving data across to a new environment. Data can be replicated over the network, it can be copied over the network, or it can be transferred using a intermediate storage as a disk or tape.

Figure 5-8 on page 77 illustrates how Portable Data Containers compare to alternate migration transports. Note that this figure is not as accurate as it could be. For example, we have not taken into account the time necessary if you transport files via tape, the load time for the tape, the transport time, or the time it takes to write the data into the new file system. If we would, then the advantage for Portable Data Containers - would have been even greater.

To calculate the throughput of Portable Data Containers, we have taken a *free* approach. Portable Data Containers, as explained earlier, do not have a throughput because they do not touch the data. So, to calculate a number, we took the total amount of data in the file system that was migrated and divided by the amount of time it took to migrate.

Table 5-4 on page 77 illustrates the different throughput values achieved with Portable Data Containers in this book.

Table 5-4 Portable Data Containers - throughput values

Number of files (file size)	Amount of data	Time to convert (seconds)	Throughput (MB/s)
1000 (1 MB)	1 GB	20.1	50
1000 (10 MB)	10 GB	19.7	508
1000 (100 MB)	100 GB	26.6	3,759

It is obvious that these values would be very hard to achieve in real life because the higher numbers are higher than the available bandwidth between memory and disk. However, this further shows the strength of Portable Data Containers and the value they add to migration.

To put it in perspective, we used the middle value from Table 5-4 and compared it to common technologies used in today's data centers (Figure 5-8).

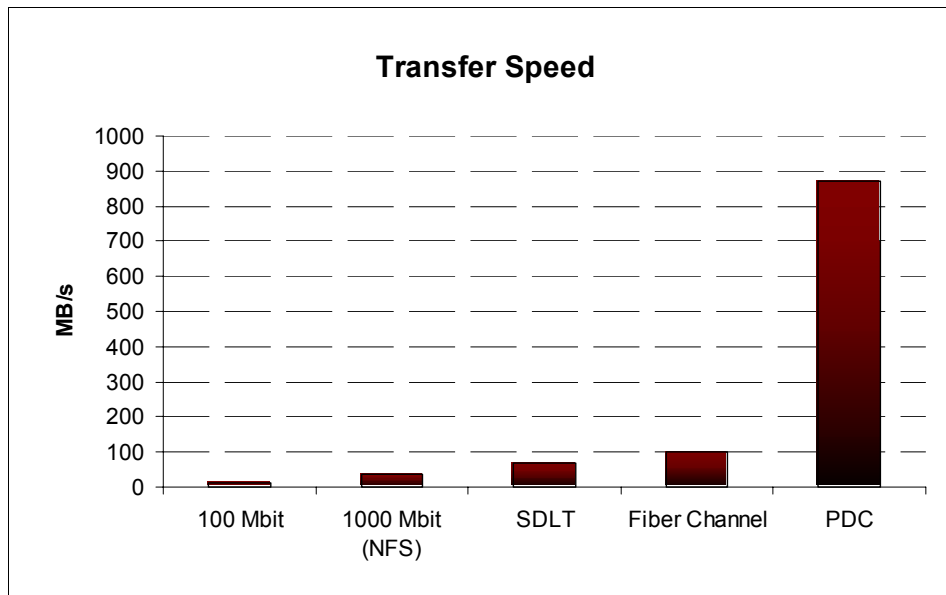


Figure 5-8 Performance comparison

## 5.5 Other performance considerations

After you have completed your Solaris to Linux migration, you should consider tuning your Linux environment for optimal performance running on IBM @server xSeries and BladeCenter hardware. To do so, review the following IBM Redbooks:

- ▶ *Tuning SUSE LINUX Enterprise Server on IBM @server xSeries Servers*, REDP-3862  
<http://publib-b.boulder.ibm.com/abstracts/redp3862.html?open>
- ▶ *Tuning Red Hat Enterprise Linux on xSeries Servers*, REDP-3861  
<http://publib-b.boulder.ibm.com/abstracts/redp3861.html?open>
- ▶ *Tuning IBM @server xSeries Servers for Performance*, SG24-5287  
<http://www-1.ibm.com/servers/eserver/linux/xseries/>

## 5.6 Conclusion

This chapter was meant to give an answer to the question: How long will it take to migrate my file system? Answering this question can be difficult. However, by using several examples, hints, and tips, you can make a very well educated estimation on how long almost any file system will take to migrate.

We have shown that the size of files does not matter as much as the amount of metadata in the file system. The number of files in a directory does not matter nor does the number of directories versus files. A directory metadata entry takes as long to convert as a file metadata entry.



## Best practices

This chapter discusses best practices for migrating data to Linux using VERITAS Portable Data Containers.

## 6.1 Minimizing data migration surprises

One of the major challenges with migration is that the data in your file system may not be compatible with the destination platform. Previously, there has not been a transparent way of migrating incompatible data, and you had to either restrict users or handle the incompatibility at the time of migration. VERITAS Portable Data Containers contain a feature specifically created to address this issue.

### 6.1.1 Preventing the file system from writing incompatible data

With Portable Data Containers, it is possible to specify with which operating system(s) the file system should be compatible. This design enables the administrator to preempt any compatibility issues by not allowing the incompatible data to be written to disk.

For example, if you are migrating data from Solaris to Linux you may run into file system size issues. VERITAS File System on Solaris is a full 64-bit file system that takes advantage of Sun's 64-bit environment, which offers a maximum file system size that is limited by the amount of storage that is attached to the server. Linux is limited to a 2 TB maximum file system size. Thus, it is impossible to migrate anything larger than 2 TBs to Linux.

A perhaps more common limitation is the user ID limitation. Today, Linux is limited to 65 536 possible user IDs, while Solaris can have as many as 2 000 000 user IDs.

The **fscdstask** utility checks or enforces the following attributes:

- ▶ File system size
- ▶ File size
- ▶ File name length
- ▶ Directory depth length
- ▶ User ID and group ID

You can also use the **fscdstask** command to validate that a file system is conforming to a specific operating system environment. The **fscdstask validate** command traverses through the file system and makes sure that all the metadata is within the specified parameters. Keep in mind, however, that this procedure may cause extra I/O load because it has to check every file in the file system.

## 6.2 Making the conversion process safer and faster

To make a data migration as safe as possible, Portable Data Containers save a copy of the metadata before the actual conversion. By saving a copy of the metadata, you can recover the file system without actually having to recover a backup if there is an unexpected error (such as a hardware failure, power outage, etc.). You can achieve this backup by rewriting the metadata that has been saved to the recovery file and doing a subsequent file system check.

While the metadata file can be a big time saver in the event of a failure, it can also become a bottleneck in the time it takes to do a conversion. If the recovery file is being stored in an NFS mount that is significantly slower than a local disk, then the conversion may be longer.

We highly recommend that you store the metadata in a file system that allows very fast write access to the disk. The key is low latency for the writes rather than high bandwidth.

The performance examples in Chapter 5, “Performance” on page 67 used a locally attached 10 000 rpm SCSI disk to store the recovery file, which was fast enough not to be a bottleneck in the conversion process.

## 6.3 Calculating the space requirements of the conversion process

Even though Portable Data Containers convert the metadata in place, they do require temporary space to store the old metadata information, the recovery file. It is not possible to predict the size of this file because of the structure of the file system and how much the file system is fragmented.

A general rule of thumb is that the more files and directories the file system contains and the more fragmented it is, the more space the recovery file will take.

The size of data files is not as important. The difference in the amount of metadata from a 10 MB file to a 1 GB file is negligible in conversion time (unless the file system is severely fragmented).

Table 6-1 on page 82 documents the space requirements during the conversion examples throughout this redbook. It should give a hint of how much space is required for the conversion.

Table 6-1 Space requirements of metadata recovery file

Number of files	Directories	File size	Recovery file size
200	20	100 MB	700 MB
2000	200	100 MB	717 MB
4,000	400	100 MB	736 MB
400	1	1 GB	701 MB
100 000	100	10 KB	727 MB
500 000	500	10 KB	843 MB
1 000 000	1000	10 KB	991 MB
5 000 000	5000	10 KB	2.15 GB
10 000 000	10 000	10 KB	3.6 GB

## 6.4 Unconverting a file system

There are instances where unconverting a file system may be necessary. Portable Data Containers technology is currently only designed to convert from the existing endian platform (the platform that is currently in use) to the destination endian platform.

For example, if the platform is Linux, Portable Data Containers can convert little endian to big endian, but not the other way around. To convert a file system from big endian to little endian, you must run the **fscdsconv** command on the big endian platform.

This situation is where the recovery file is very useful. Because the recovery file contains all the metadata from before the actual conversion, you can use it to re-apply the metadata and undo the entire conversion process.

You can use the **fscdsconv** command to apply the recovery file to a previously converted file system. It is important that absolutely nothing has changed in the file system because any changes may be lost.

```
fscdsconv -r -f <recovery file> <special device>
```



## 6.5 Using VERITAS FlashSnap for a test migration

Use VERITAS FlashSnap to do a test migration or to migrate when your application cannot be shut down but can be put in read-only mode during the actual migration.

### 6.5.1 Requirements and recommendations

This section lists the requirements and recommendations for using FlashSnap.

#### SAN Connection environment

VERITAS Storage Foundation requires a SAN connection environment using most of its functions. It should be verified to connect servers and storage subsystem correctly.

#### FAStT Storage Manager storage partitioning feature

FAStT requires the use of its storage partitioning feature for making multiple partitions. After splitting the diskgroup, you must remap the logical drives of storage subsystem to different host.

To verify that the storage partitioning feature has been installed, you should use the TotalStorage® FAStT Storage Manager. To do so, click **Storage Subsystem** → **Premium Feature** → **List** in the Subsystem Management window to display the List Premium Features window (Figure 6-1).

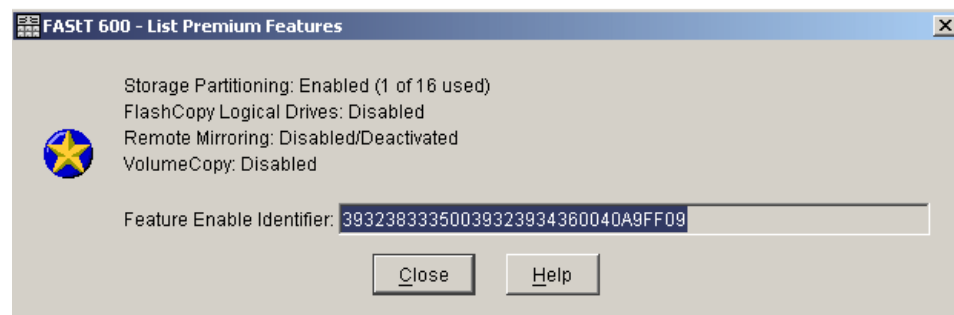
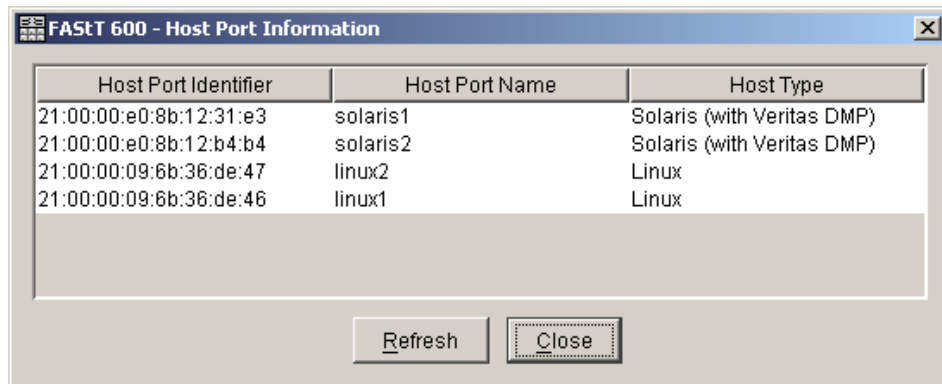


Figure 6-1 List of premium features

After the feature has been installed, FASTT is enabled to make multiple host mapping and multiple host types (Figure 6-2).



Host Port Identifier	Host Port Name	Host Type
21:00:00:e0:8b:12:31:e3	solaris1	Solaris (with Veritas DMP)
21:00:00:e0:8b:12:b4:b4	solaris2	Solaris (with Veritas DMP)
21:00:00:09:6b:36:de:47	linux2	Linux
21:00:00:09:6b:36:de:46	linux1	Linux

Figure 6-2 Multiple host type definitions

## Device driver

SAN connection requires the latest version of host bus adapter (HBA) device driver on Linux. Verify that the version of the HBA device driver matches the Linux kernel. The supported kernel version of HBA device driver is listed in its README file. Read it carefully before using the HBA device driver.

## Disks

VERITAS FlashSnap requires two disks belonging to one diskgroup at least, one for original volume and the other for snapshot volume. That means a dedicated disk is needed for each of original volume and snapshot volume.

## Array Support Library

If you use Dynamic Multi Pathing, VERITAS Volume Manager requires non-failover Fibre Channel driver and Array Support Library (ASL). ASL is provided by VERITAS Certified Storage program. You can download it from VERITAS support site.

## VERITAS Enterprise Administrator

VERITAS Enterprise Administrator (VEA) has a GUI and provides the efficiency of VERITAS Volume Manager and FlashSnap operation. VEA can access VERITAS Volume Manager remotely via network. It can also manage all the servers which VERITAS Volume Manager installed on network.

VEA is not required but recommended for efficiency of VERITAS Volume Manager. Of course, command operation is also provided. For more information

about VEA, see *VERITAS Volume Manager User's Guide - VERITAS Enterprise Administrator* that comes with the product software.

## 6.5.2 Migration example environment

Figure 6-3 illustrates the environment of the migration example.

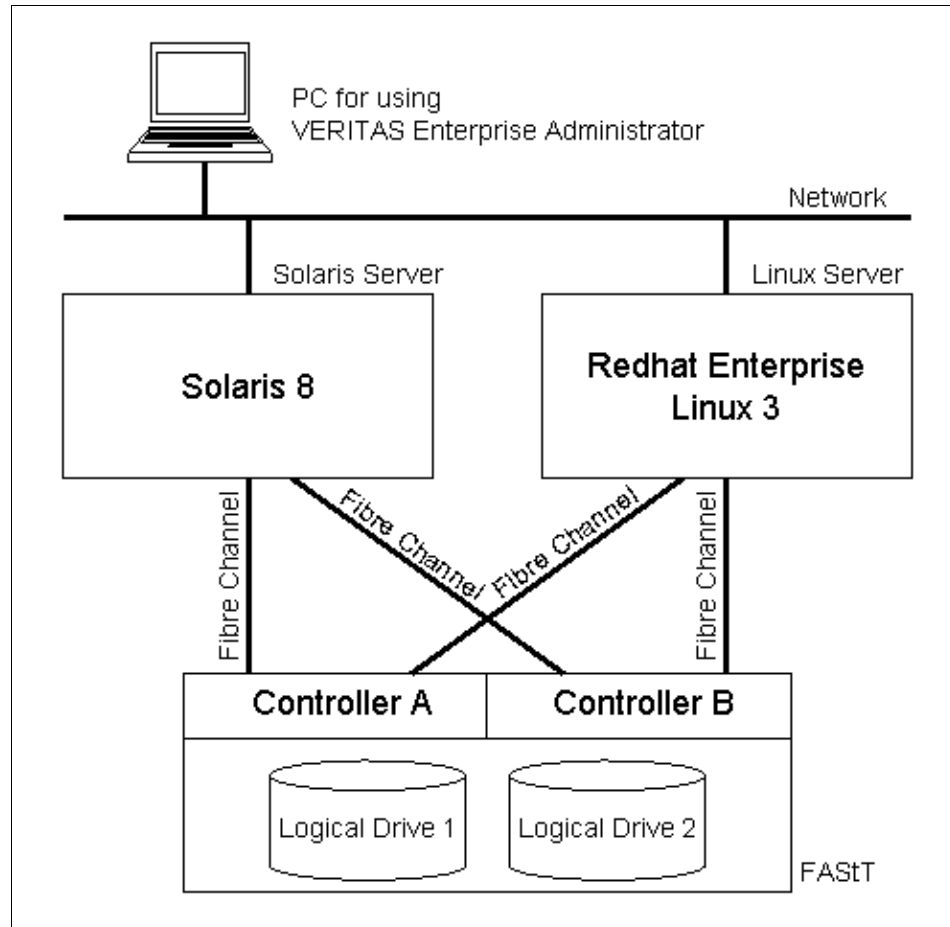


Figure 6-3 Migration test environment

At first, a shared storage subsystem is required. The storage subsystem is connected to each server using Fibre Channel, and the connection is a dual path. The connection using dual path is not required but recommended for high availability. There are two logical drives on the storage subsystem, one is for the volume of original data and one for making a snapshot volume.

Network connection is only used for the operation of VERITAS Volume Manager using VERITAS Enterprise Administrator.

Of course, VERITAS Storage Foundation 4.0 should be installed on each of Solaris and Linux.

### 6.5.3 Situation before migration

This section describes the situation before migration.

#### FAStT configuration

Before migration, you should consider a logical drive mapping configuration of FAStT. This configuration is needed to verify the logical drive mapping for Solaris and Linux as shown in Figure 6-4.

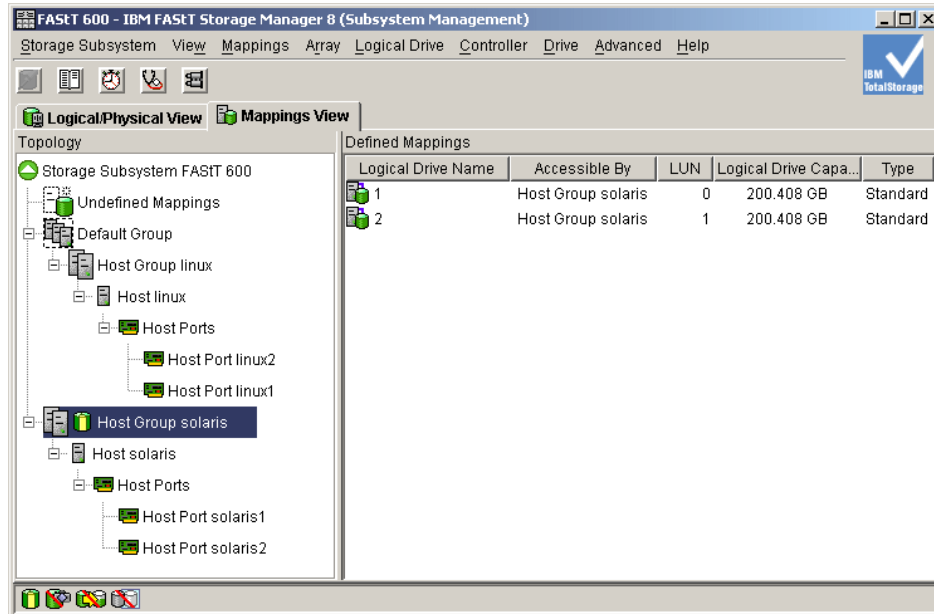


Figure 6-4 Mapping configuration before migration

All logical drives (1 and 2) are mapped to Solaris, and nothing is mapped to Linux.

## Diskgroup and volume

All of diskgroup and volume belong to the Solaris server. For example, the original diskgroup is named *origdg*, and the original volume is named *origvol*.

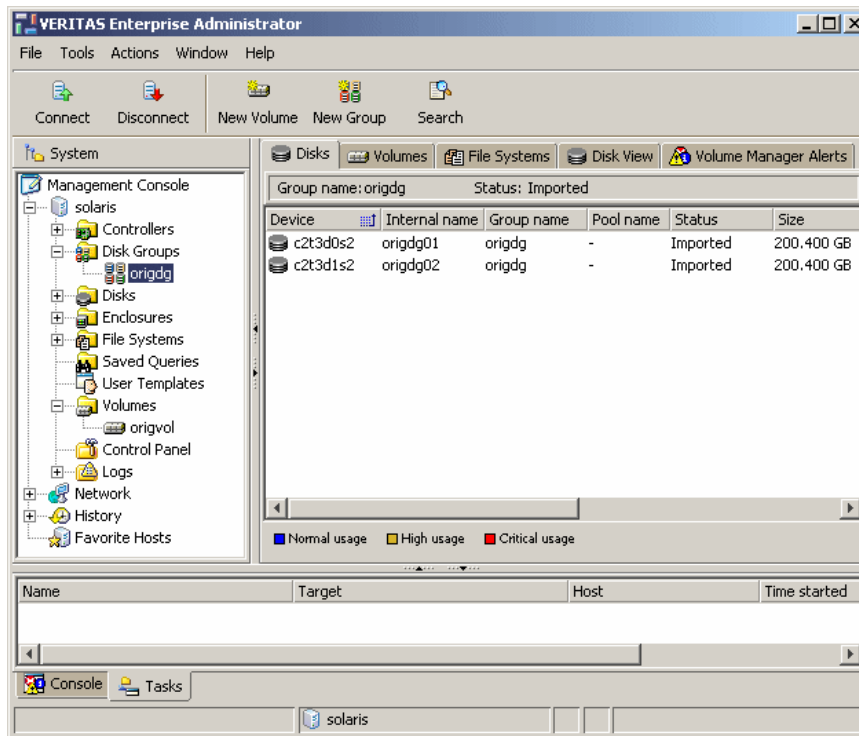


Figure 6-5 Diskgroup and volume configuration before migration

Disk *origdg01* corresponds to *logical drive 1*, and disk *origdg02* corresponds to *logical drive 2* on FASTT. Two disks should belong to the same diskgroup, *origdg*, to create the snapshot volume.

In this case, disk *origdg01* is assigned to the original volume *origdg*. The other disk should be reserved for creating the snapshot volume.

## 6.5.4 Summary of migration steps

To migrate using Portable Data Containers and FlashSnap, do the following:

1. Verify the diskgroup, which is made from two disks (Figure 6-6).

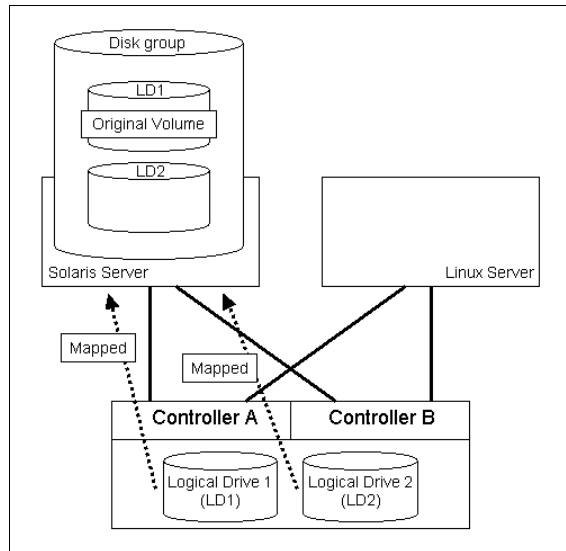


Figure 6-6 Initial configuration before migration

2. Create a snapshot volume using a dedicated disk in the same disk group on Solaris (Figure 6-7 on page 89).

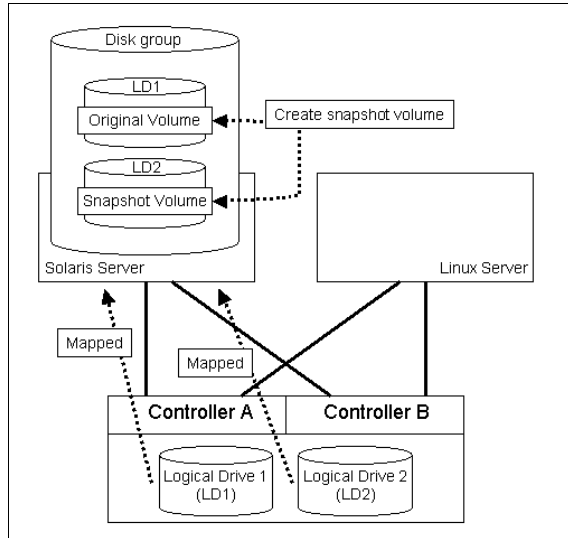


Figure 6-7 Making snapshot volume using dedicated disk

3. Split the snapshot volume from original volume on Solaris (Figure 6-8).

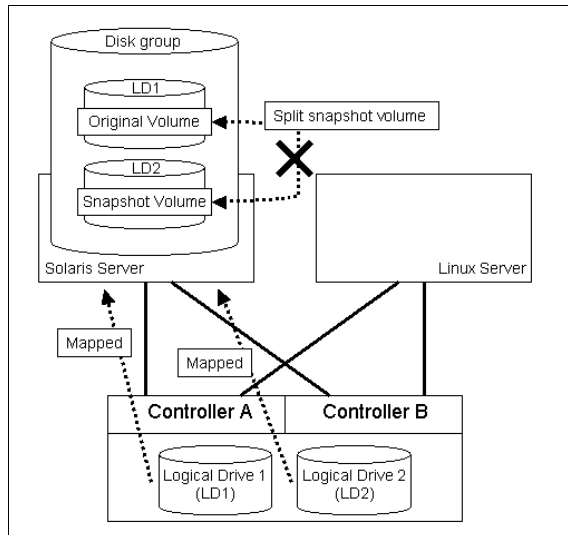


Figure 6-8 Split snapshot volume

4. Split the original diskgroup on Solaris (Figure 6-9).

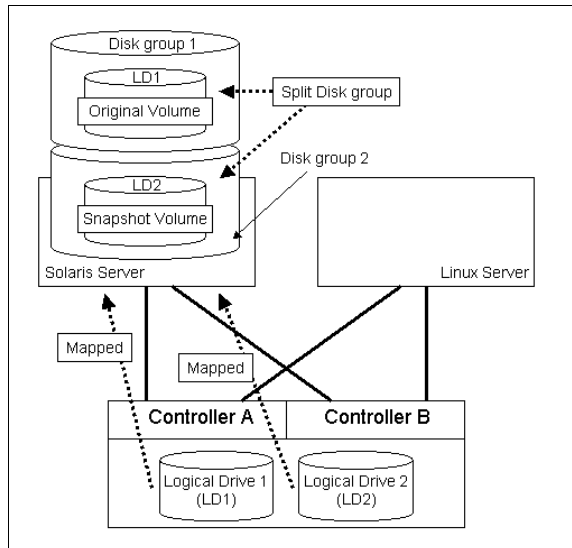


Figure 6-9 Split diskgroup

5. Execute the `fscdconv` command for the snapshot volume on Solaris (Figure 6-10).

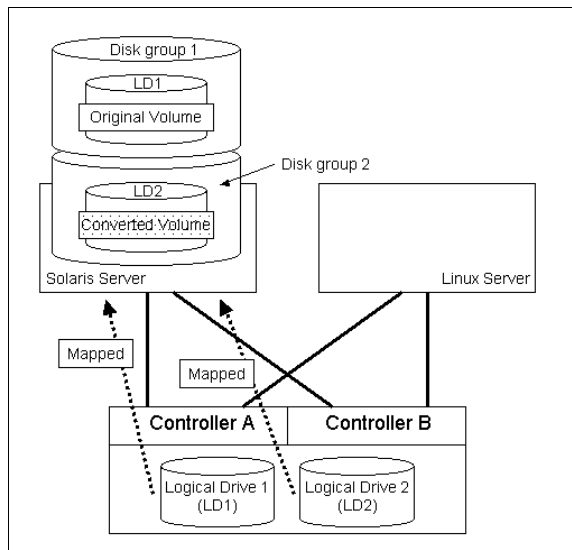


Figure 6-10 Conversion file system on snapshot volume



6. Deport the diskgroup which includes the snapshot volume from Solaris.
7. Change the logical drive mapping of the logical drive which is used by the snapshot volume on the storage subsystem.
8. Import the diskgroup to Linux.

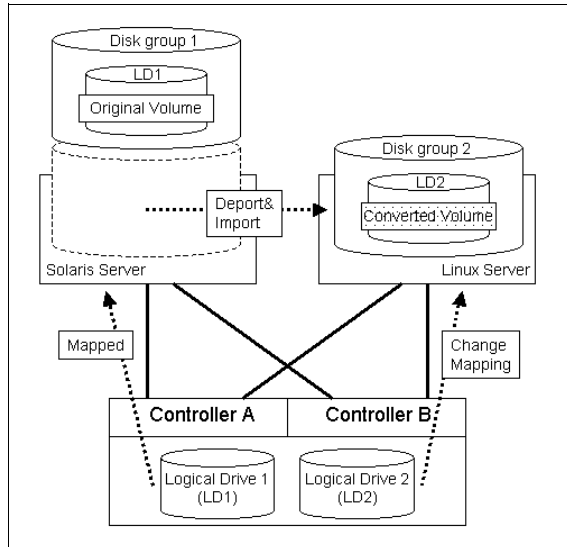


Figure 6-11 Deporting and importing disk group to Linux

9. After importing the diskgroup, mount the file system on Linux.

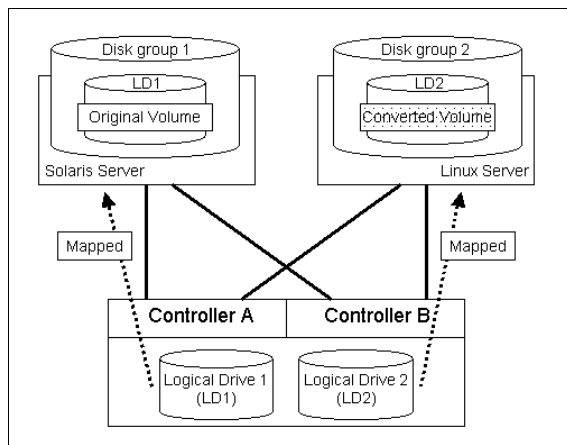


Figure 6-12 Configuration after migration

The most important step of this migration is the file system conversion using the **fscdconv** command, the part of Portable Data Containers. The conversion

makes data migration possible without data transfer between Solaris and Linux. FlashSnap is also a useful function for migration. It makes data copy possible without the influence of the original data. Portable Data Containers and FlashSnap work with great efficiency for migration.

## 6.5.5 Migration example steps

This section describes the migration steps using Portable Data Containers and FlashSnap. It is better to use VERITAS Enterprise Administrator (VEA) for easy operations.

### Create and split snapshot

At first, you are required to create a snapshot volume for migrating data. The snapshot volume is a copy of the complete original volume. For example, the snapshot volume is named as *migvol*.

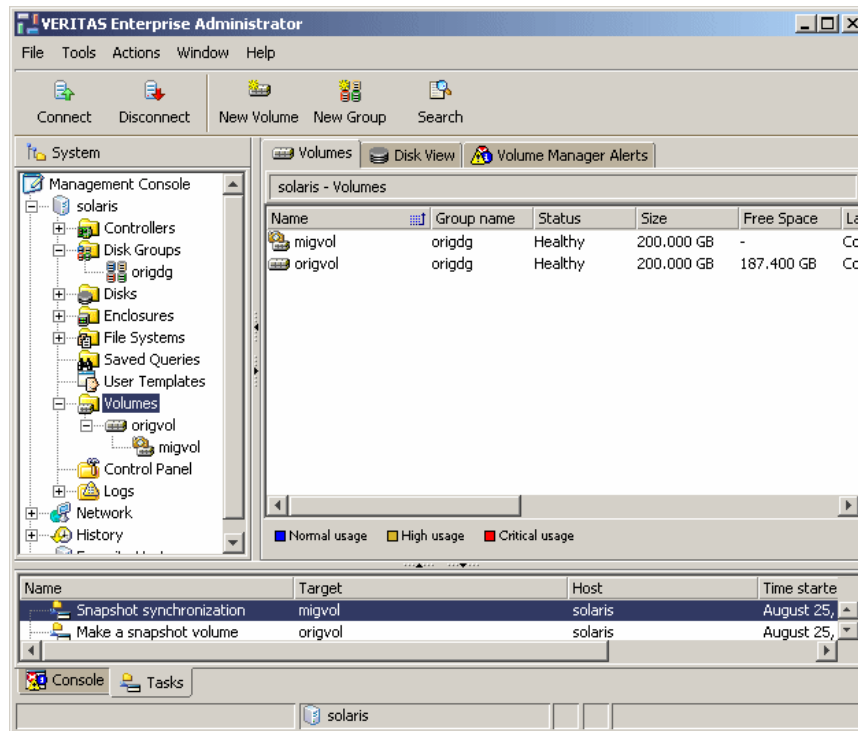


Figure 6-13 Created snapshot volume

You need to assign disk(s) explicitly because the snapshot volume should be created on the dedicated disk in the same diskgroup. The original volume and

the snapshot volume are synchronized as default. At the first creation of the snapshot volume, it will take some time depending on the volume size.

You can also use a mirrored volume to reattach the snapshot volume to the original volume. If the snapshot volume is needed, it can be a visible volume as snapshot to break off the mirror volume.

Next, you need to split the created snapshot volume from the original volume (Figure 6-14). Afterwards, the snapshot volume *migvol* becomes an independent volume.

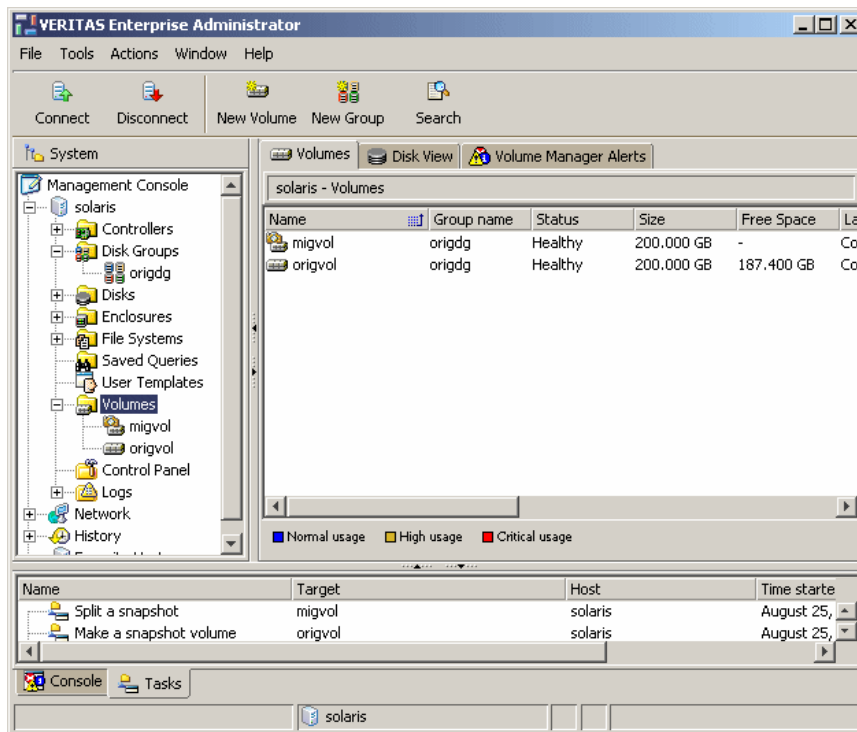


Figure 6-14 Split snapshot volume

At this moment, there is no relation between the original volume and snapshot volume.

During this work, the original volume does not need to be unmounted. Thus, if some kind of service is using the original volume on Solaris, you should not stop the service for this work.

## Split diskgroup

After splitting the snapshot volume, you should split the diskgroup (Figure 6-15). For example, the split diskgroup is named as *migdg*.

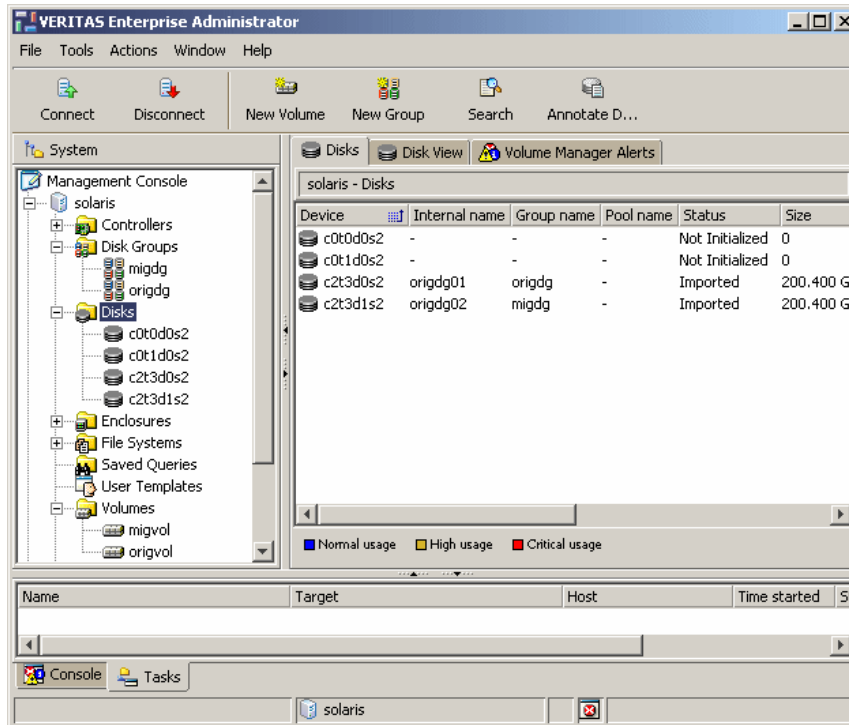
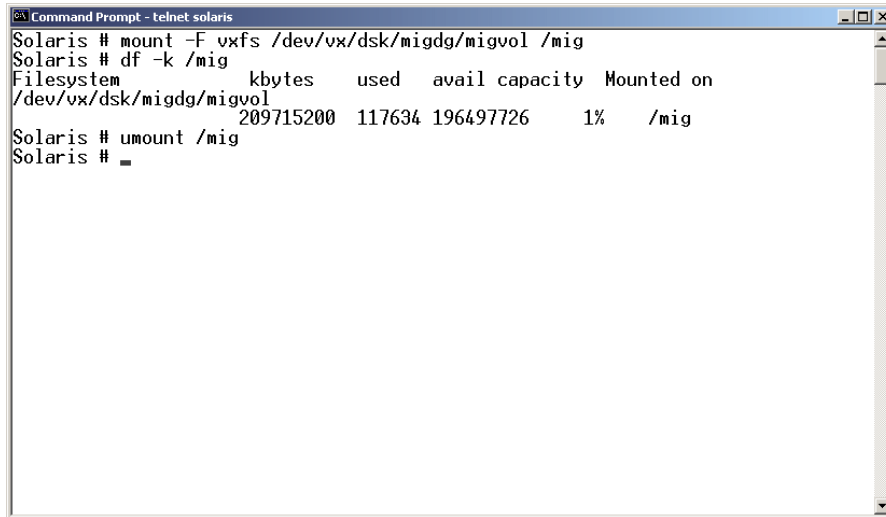


Figure 6-15 Split diskgroup

The split diskgroup *migdg* includes the disk *origdg02* which the volume *migvol* belongs to. You should use it to split the disk for migration using FlashSnap. Keep in mind which disk the volume uses and to which diskgroup the disk belongs.

## File system conversion

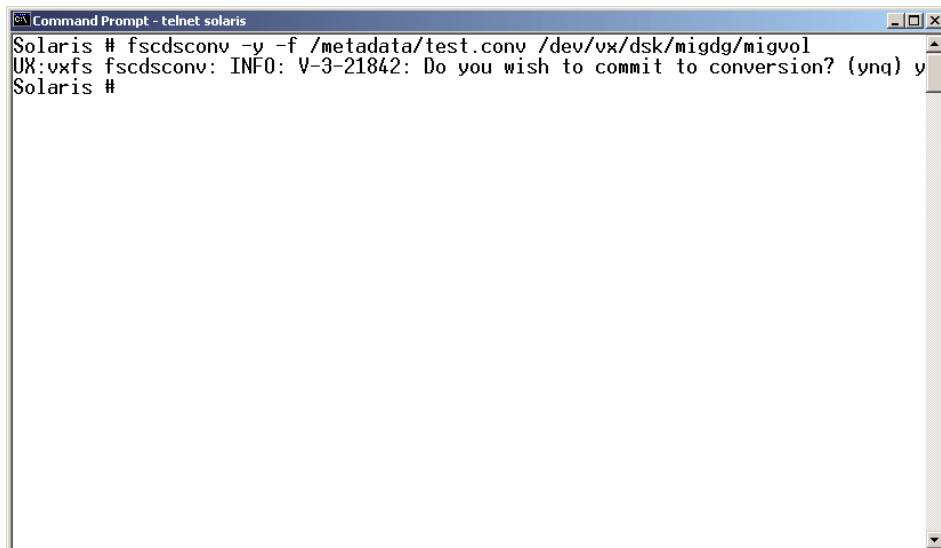
At first, you are required to mount and unmount file system of the migrated volume (Figure 6-16).



```
Command Prompt - telnet solaris
Solaris # mount -F vxfs /dev/vx/dsk/migdg/migvol /mig
Solaris # df -k /mig
Filesystem          kbytes    used   avail capacity  Mounted on
/dev/vx/dsk/migdg/migvol
                    209715200 117634 196497726    1%    /mig
Solaris # umount /mig
Solaris #
```

Figure 6-16 Mount and unmount file system

After you unmount file system, execute the **fscdsconv** command for file system conversion as shown in Figure 6-17.



```
Command Prompt - telnet solaris
Solaris # fscdsconv -y -f /metadata/test.conv /dev/vx/dsk/migdg/migvol
UX:vxfs fscdsconv: INFO: V-3-21842: Do you wish to commit to conversion? (ynq) y
Solaris #
```

Figure 6-17 File system conversion using fscdsconv

## Deport diskgroup from Solaris

After file system conversion, you are required to deport the migrated diskgroup from Solaris. When the diskgroup is deported, the ID and server name of original owner are added to the diskgroup name. Its status is also changed to Deported as shown in Figure 6-18.

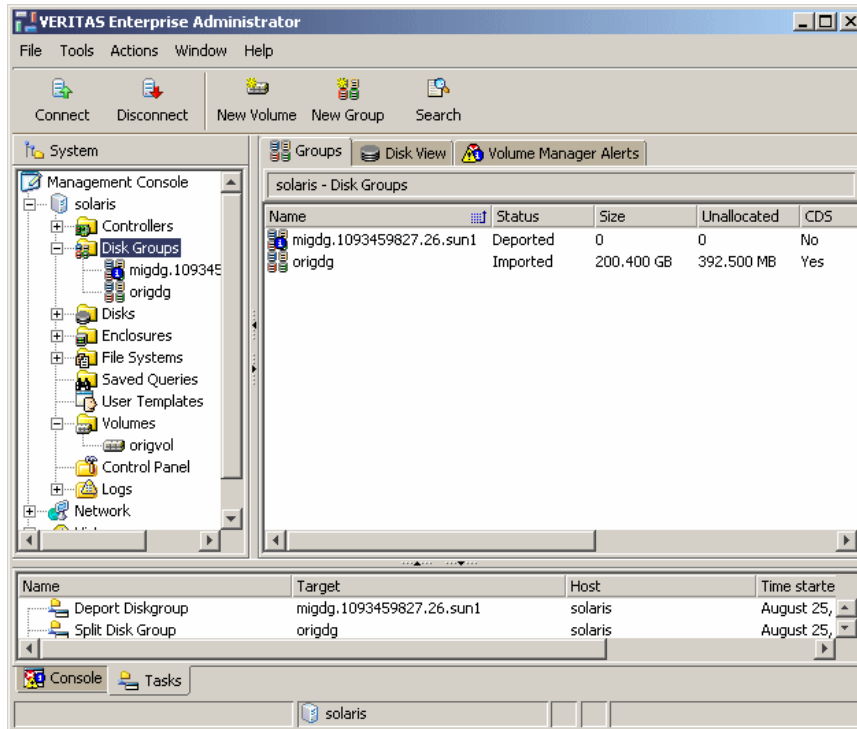


Figure 6-18 Deported diskgroup

## Change logical drive mapping on FASTT

You are required to change the logical drive mapping of the FASTT for migrating data from Solaris to Linux. *Logical Drive 2* is assigned to the disk of the snapshot volume. It should be remapped to the Linux host group as shown in Figure 6-19 on page 97.

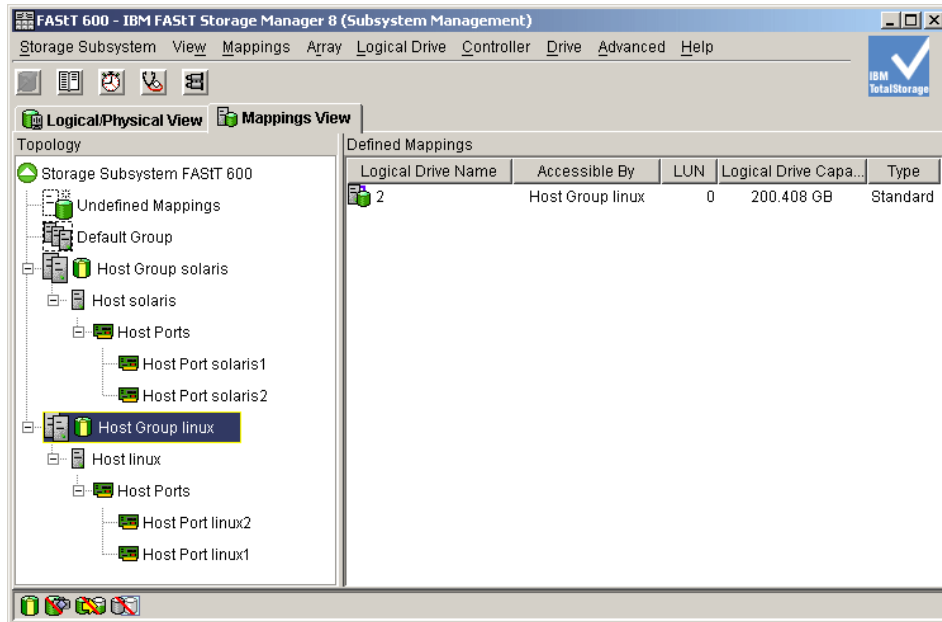


Figure 6-19 Change host mapping

After changing the mapping, reboot the Linux server so that it recognizes the new host mapping. Do not reboot the Solaris server. Instead, use it to execute **Rescan** of VEA to recognize the new host mapping.

The VERITAS Volume Manager on Linux should see the diskgroup after rebooting (Figure 6-20 on page 98).

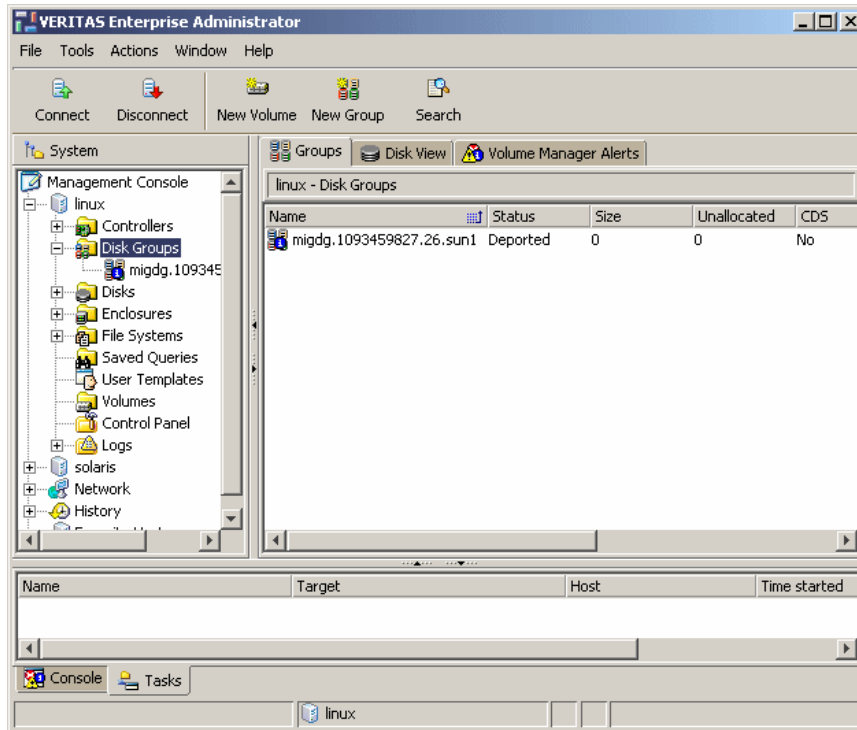


Figure 6-20 Diskgroup on Linux right after reboot

The diskgroup is deported on the Linux server immediately after reboot. It cannot be used for migration in this status.

### Import diskgroup to Linux

You should import the diskgroup to use the diskgroup that has been migrated from Solaris.



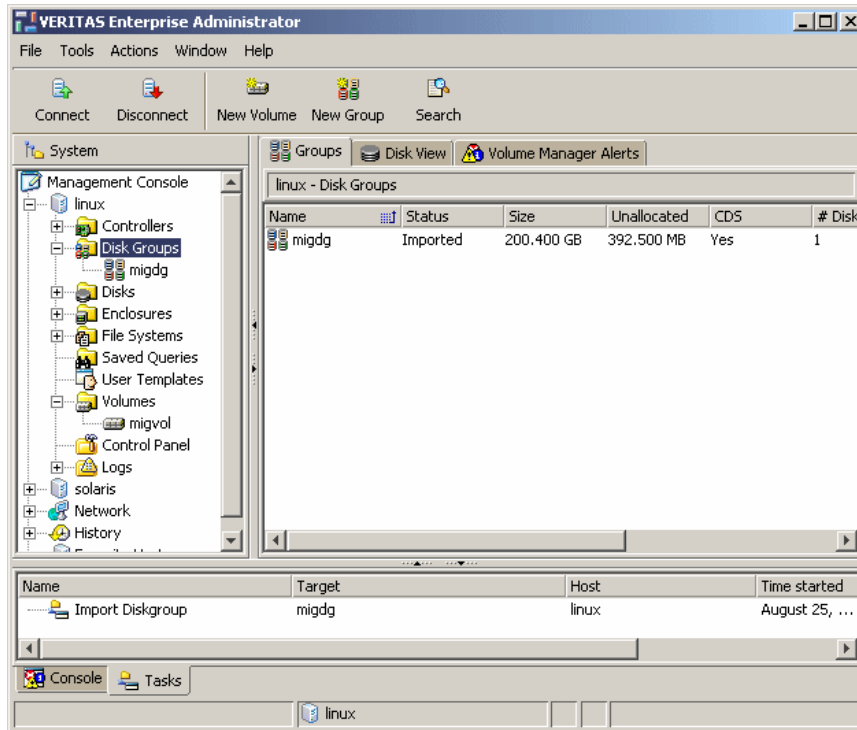
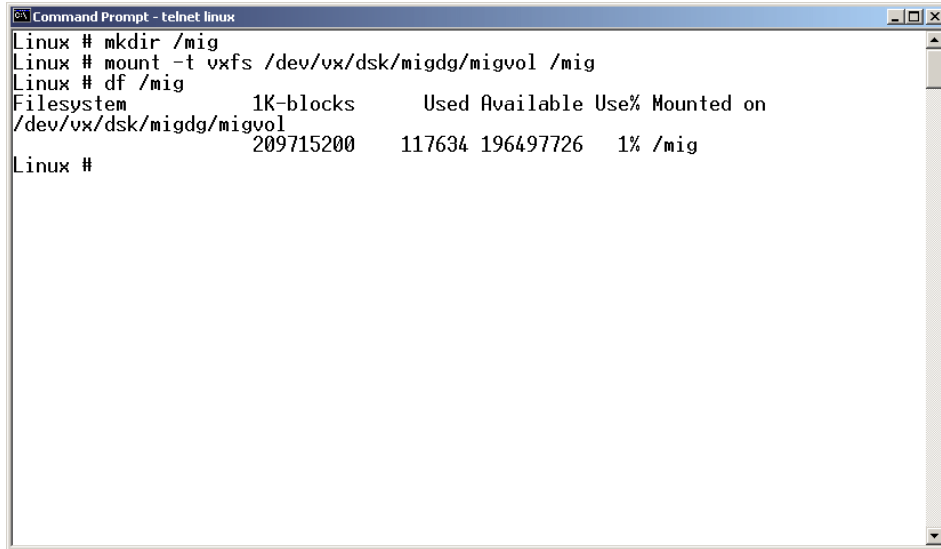


Figure 6-21 Import diskgroup on Linux

At this moment, you can use the volume contained in the diskgroup. In this example, the imported diskgroup is *migdg*, and the volume is *migvol*.

### Mount file system on Linux

Finally, mount the file system of the imported diskgroup on Linux. In this example, the mount point is created as */mig* as shown in Figure 6-22 on page 100.



```
Command Prompt - telnet linux
Linux # mkdir /mig
Linux # mount -t vxfs /dev/vx/dsk/migdg/migvol /mig
Linux # df /mig
Filesystem            1K-blocks      Used Available Use% Mounted on
/dev/vx/dsk/migdg/migvol
                        209715200    117634 196497726   1% /mig
Linux #
```

Figure 6-22 Mount file system on Linux

Of course, the mount point depends on the object of the migration, such as an application. The mount point should be the same as the one on Solaris.

## 6.6 Oracle application migration scenario

This section demonstrates a real-world application migration scenario. Let's consider migrating an Oracle application running on Solaris 8 to Linux AS 3.0. The source side is Oracle version 8.1.7 and the target side is Oracle version 9.2.0.4.0.

**Note:** Before performing any migration of an Oracle database, you should consult the document *Compatibility Matrix for Export and Import Between Different Oracle Versions* that is available at:

<http://metalink.oracle.com>

This document will help you to determine whether the scenario you are performing is a supported case of migration.

We should mention that it is possible to perform an online migration as proposed in the example 6.5.5, "Migration example steps" on page 92. However, the migration demonstration illustrated here restricts itself to VERITAS Storage Foundation 4.0 Portable Data Containers. We have skipped the online migration

part which can be achieved by breaking off of a mirror and splitdg operation to stay focused on migration using Portable Data Containers.

Before you begin the migration, you must install Oracle 8.1.7 on Solaris 8 and Oracle 9.2.0.4.0 on Linux Red Hat AS 3.0. You can use the documentation that comes with each respective software package to install Oracle. It is very important to follow these instructions.

To migrate the system, you will need to follow these steps:

1. Create a sample database on source side Oracle.
2. Export the database using the exp utility of Oracle 8.1.7.
3. Perform the Portable Data Containers conversion step on the Solaris side by unmounting the VERITAS file system and deporting the disk group on the Solaris side.
4. Import the disk group and mount the VERITAS file system on the Linux side. Ensure that pre-import procedures are followed on the Oracle server on the Linux side.
5. Perform an Oracle application import procedure on Linux.
6. Verify the migration. A successful migration happens if same the Oracle servers on the source and target platform respond with an identical response.

The sections that follow provide details on how to perform these steps.

### **Step 1: Create a sample database on source side Oracle**

Logon as Oracle user on Solaris side. Reach the svrmgr1 prompt by running the **svrmgr1** command. At the svrmgr1 prompt, type `connect internal`. Run the SQL script in Example 6-1. The SQL script creates an initial database.

*Example 6-1 Post installation database creation on Oracle 8.1.7 on Solaris 8*

---

```
create database
controlfile reuse
logfile
'~/log_bench_1' size 100M REUSE,
'~/log_bench_2' size 100M REUSE
datafile
'~/dbs/bench.dbf' size 400M REUSE
maxinstances 1
maxdatafiles 1022
maxlogfiles 36
archiveolog
/
create rollback segment bootstrap storage(initial 100k next 100k)
/
```

```
alter rollback segment bootstrap online
/  
create tablespace tbs1 datafile '?/dbs/usrl.f' size 500M reuse;
```

---

For preparing the correct catalogs required for exporting the database, run the following scripts, which are available in the standard Oracle home:

```
@?/rdbms/admin/catalog  
@?/rdbms/admin/catproc
```

From the svrmgrl prompt, run the SQL script in Example 6-2. A sample table EMP and tablespace is created.

*Example 6-2 Listing of script for sample database creation on oracle 8.1.7*

---

```
connect system/manager  
create rollback segment rbs_1 storage(initial 100k next 100k)  
tablespace tbs1;  
alter rollback segment rbs_1 online;  
create table emp (id number, name char(1024), addr varchar(2048))  
tablespace tbs1;  
  
insert into emp values (1, 'first name last name', 'One Universal Dr.');
```

```
insert into emp select * from emp;  
insert into emp select * from emp;  
insert into emp select * from emp;  
insert into emp select * from emp;  
insert into emp select * from emp;  
insert into emp select * from emp;  
insert into emp select * from emp;  
insert into emp select * from emp;  
insert into emp select * from emp;  
insert into emp select * from emp;  
insert into emp select * from emp;  
insert into emp select * from emp;  
insert into emp select * from emp;  
insert into emp select * from emp;  
insert into emp select * from emp;  
insert into emp select * from emp;
```

---

## Step 2: Export the database

Run the commands highlighted in blue in Example 6-3 to export the database from Solaris with a dump file created on a VERITAS File System and Volume Manager stack.

### Example 6-3 Exporting of the database on oracle 8.1.7 running on Solaris

---

```
Script started on Wed 15 Sep 2004 11:30:26 AM EDT
Solaris # clear
Solaris # which exp
//ora17/oracle/product/8.1.7/bin/exp
Solaris # hostname
sun1
Solaris # uname -a
SunOS sun1 5.8 Generic_117350-06 sun4u sparc SUNW,Ultra-60
Solaris # set |grep ORACLE
ORACLE_BASE=/ora17/oracle
ORACLE_HOME=/ora17/oracle/product/8.1.7
ORACLE_SID=testdb
Solaris # exp system/manager FILE=/exp/ora8i.exp GRANTS=y ROWS=y
```

Export: Release 8.1.7.0.0 - Production on Wed Sep 15 11:32:10 2004

(c) Copyright 2000 Oracle Corporation. All rights reserved.

```
Connected to: Oracle8i Enterprise Edition Release 8.1.7.0.0 - 64bit Production
With the Partitioning option
JServer Release 8.1.7.0.0 - 64bit Production
Export done in US7ASCII character set and US7ASCII NCHAR character set
```

About to export specified users ...

- . exporting pre-schema procedural objects and actions
- . exporting foreign function library names for user SYSTEM
- . exporting object type definitions for user SYSTEM

About to export SYSTEM's objects ...

- . exporting database links
- . exporting sequence numbers
- . exporting cluster definitions
- . about to export SYSTEM's tables via Conventional Path ...
  - . . exporting table DEF\$\_CALLDEST 0 rows exported
  - . . exporting table DEF\$\_DEFAULTDEST 0 rows exported
  - . . exporting table DEF\$\_DESTINATION 0 rows exported
  - . . exporting table DEF\$\_ERROR 0 rows exported
  - . . exporting table DEF\$\_LOB 0 rows exported
  - . . exporting table DEF\$\_ORIGIN 0 rows exported
  - . . exporting table DEF\$\_PROPAGATOR 0 rows exported
  - . . exporting table DEF\$\_PUSHED\_TRANSACTIONS 0 rows exported

```

. . exporting table          DEF$_TEMP$LOB          0 rows exported
. . exporting table          EMP                    16384 rows exported
. . exporting table          SQLPLUS_PRODUCT_PROFILE 0 rows exported
. exporting synonyms
. exporting views
. exporting stored procedures
. exporting operators
. exporting referential integrity constraints
. exporting triggers
. exporting indextypes
. exporting bitmap, functional and extensible indexes
. exporting posttables actions
. exporting snapshots
. exporting snapshot logs
. exporting job queues
. exporting refresh groups and children
. exporting dimensions
. exporting post-schema procedural objects and actions
. exporting statistics

```

Export terminated successfully without warnings.

```
Solaris # mount |grep exp
```

```
/exp on /dev/vx/dsk/oracledg/exp read/write/setuid/delaylog/largefiles/ioerror=
wdisable/dev=3d0232b on Mon Sep 13 15:18:10 2004
```

```
Solaris # vxprint -ht
```


```
Disk group: oracledg
```

DG NAME	NCONFIG	NLOG	MINORS	GROUP-ID			
ST NAME	STATE	DM_CNT	SPARE_CNT	APPVOL_CNT			
DM NAME	DEVICE	TYPE	PRIVLEN	PUBLEN	STATE		
RV NAME	RLINK_CNT	KSTATE	STATE	PRIMARY	DATAVOLS	SRL	
RL NAME	RVG	KSTATE	STATE	REM_HOST	REM_DG	REM_RLNK	
CO NAME	CACHEVOL	KSTATE	STATE				
VT NAME	NVOLUME	KSTATE	STATE				
V NAME	RVG/VSET/CO	KSTATE	STATE	LENGTH	READPOL	PREFPLEX	UTYPE
PL NAME	VOLUME	KSTATE	STATE	LENGTH	LAYOUT	NCOL/WID	MODE
SD NAME	PLEX	DISK	DISKOFFS	LENGTH	[COL/]OFF	DEVICE	MODE
SV NAME	PLEX	VOLNAME	NVOLLAYR	LENGTH	[COL/]OFF	AM/NM	MODE
SC NAME	PLEX	CACHE	DISKOFFS	LENGTH	[COL/]OFF	DEVICE	MODE
DC NAME	PARENTVOL	LOGVOL					
SP NAME	SNAPVOL	DCO					
dg oracledg	default	default	9000	1095102646.12.sun1			
dm oracledg01	c2t3d0s2	auto	2048	420263680			
v exp	-	ENABLED	ACTIVE	20971520	SELECT	-	fsgen
pl exp-01	exp	ENABLED	ACTIVE	20971520	CONCAT	-	RW

```

sd oracledg01-04 exp-01      oracledg01 62914560 20971520 0      c2t3d0  ENA
v  oracle0      -            ENABLED  ACTIVE  20971520 SELECT  -        fsgen
pl oracle0-01   oracle0      ENABLED  ACTIVE  20971520 CONCAT  -        RW
sd oracledg01-01 oracle0-01  oracledg01 0      20971520 0      c2t3d0  ENA
v  oracle1      -            ENABLED  ACTIVE  20971520 SELECT  -        fsgen
pl oracle1-01   oracle1      ENABLED  ACTIVE  20971520 CONCAT  -        RW
sd oracledg01-02 oracle1-01  oracledg01 20971520 20971520 0      c2t3d0  ENA
v  oracle2      -            ENABLED  ACTIVE  20971520 SELECT  -        fsgen
pl oracle2-01   oracle2      ENABLED  ACTIVE  20971520 CONCAT  -        RW
sd oracledg01-03 oracle2-01  oracledg01 41943040 20971520 0      c2t3d0  ENA
Solaris # exit
exit

```



script done on Wed 15 Sep 2004 11:34:34 AM EDT

### Step 3: Perform the Portable Data Containers conversion step on the Solaris side

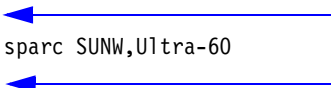
Run the commands highlighted in blue in Example 6-4 to perform the conversion of the VERITAS File System and Volume Manager stack to be accessible from the Linux side. This involves unmounting of the VERITAS File System, namely the /exp in the present case, and running the **fscdsconv** command on the file system. Note that the file system and volume manager operation is done after logging in as the root on the Solaris machine.

#### Example 6-4 VERITAS File System and Volume Manager conversion

```

Script started on Wed 15 Sep 2004 11:45:47 AM EDT
Solaris # uname -a
SunOS sun1 5.8 Generic_117350-06 sun4u sparc SUNW,Ultra-60
Solaris # pkginfo -l VRTSvxfs
  PKGINST: VRTSvxfs
    NAME: VERITAS File System
  CATEGORY: system,utilities
    ARCH: sparc
  VERSION: 4.0,REV=GA04
  BASEDIR: /
  VENDOR: VERITAS Software

```



```

DESC: Commercial File System
PSTAMP: VERITAS-FS-4.0.1.0-2004-06-03-MP1=116688-01
INSTDATE: Aug 17 2004 11:42
HOTLINE: (800) 342-0652
EMAIL: support@veritas.com
STATUS: completely installed
FILES: 233 installed pathnames
        33 shared pathnames
        5 linked files
        48 directories
        74 executables
        5 setuid/setgid executables
        50999 blocks used (approx)

```

Solaris # `pkginfo -l VRTSvxvm` ←

```

PKGINST: VRTSvxvm
NAME: VERITAS Volume Manager, Binaries
CATEGORY: system
ARCH: sparc
VERSION: 4.0,REV=12.06.2003.01.35
BASEDIR: /
VENDOR: VERITAS Software
DESC: Virtual Disk Subsystem
PSTAMP: VERITAS-4.0R_p3.7 (MP1):2004-06-02
INSTDATE: Aug 17 2004 11:44
HOTLINE: 800-342-0652
EMAIL: support@veritas.com
STATUS: completely installed
FILES: 823 installed pathnames
        27 shared pathnames
        18 linked files
        97 directories
        411 executables
        275413 blocks used (approx)

```

Solaris # `uname-a` ←

```
SunOS sun1 5.8 Generic_117350-06 sun4u sparc SUNW,Ultra-60
```

Solaris # `mount |grep exp` ←

```
/exp on /dev/vx/dsk/oracledg/exp read/write/setuid/delaylog/largefiles/ioerror=m
wdisable/dev=3d0232b on Mon Sep 13 15:18:10 2004
```

Solaris # `df -k` ←

Filesystem	kbytes	used	avail	capacity	Mounted on
/dev/dsk/c0t0d0s0	1621767	1156361	416753	74%	/
/dev/dsk/c0t0d0s4	2056211	1154759	839766	58%	/usr
/proc	0	0	0	0%	/proc
fd	0	0	0	0%	/dev/fd
mnttab	0	0	0	0%	/etc/mnttab
swap	1242736	16	1242720	1%	/var/run
dmpfs	1242720	0	1242720	0%	/dev/vx/dmp



```

dmpfs          1242720      0 1242720    0%  /dev/vx/rdmp
swap          1243128      408 1242720    1%  /tmp
/dev/dsk/c0t1d0s5 14455287 3880222 10430513   28%  /ora17
/dev/dsk/c0t0d0s3 6196234 1775356 4358916   29%  /opt
/dev/dsk/c0t0d0s7 6467926 930389 5472858   15%  /export/home
/dev/vx/dsk/oracle1edg/exp
10485760 52877 9780835    1%  /exp
/vol/dev/dsk/c0t6d0/cfsdemonstration2
595394 595394      0 100%  /cdrom/cfsdemonstration2

```

Solaris # `ls -l /exp` ←

```

total 33684
-rw-r--r--  1 oracle  oinstall  4890 Sep 15 11:36 exp-session
drwxr-xr-x  2 oracle  oinstall   96 Sep 13 15:18 lost+found
-rw-r--r--  1 oracle  oinstall 17241088 Sep 15 11:33 ora8i.exp
-rw-r--r--  1 oracle  oinstall   0 Sep 15 11:26 testfile

```

Solaris # `umount /exp` ←

Solaris # `which fscdsconv` ←

/opt/VRTS/bin/fscdsconv

Solaris # `fscdsconv -y -f /metadata/test.conv /dev/vx/dsk/oracle1edg/exp` ←

UX:vxfs fscdsconv: INFO: V-3-21842: Do you wish to commit to conversion? (ynq) y

Solaris # `vxdg list` ←

```

NAME      STATE      ID
oracle1edg  enabled,cds 1095102646.12.sun1

```

Solaris # `vxdg deport oracle1edg` ←

Solaris # `exit` ←

script done on Wed 15 Sep 2004 02:51:10 PM EDT

---

## Step 4: Import the disk group and mount

This brings us to the midway of the migration process. Next, we detach the storage from the Solaris machine and attach the same to the Linux box. The FASTt manager is used to change the storage mapping from Solaris to Linux.

1. From the FASTt Storage Manager, select **Mappings View**.

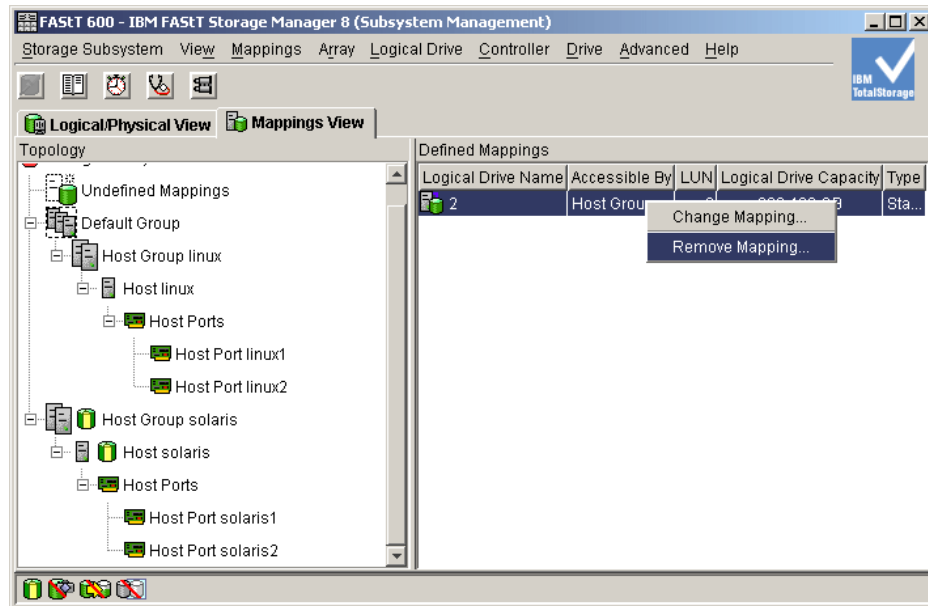


Figure 6-23 Remove mapping from Solaris

2. Right-click the Logical Drive Name. Click **Remove Mapping**.
3. At the confirmation window, select **Yes**.

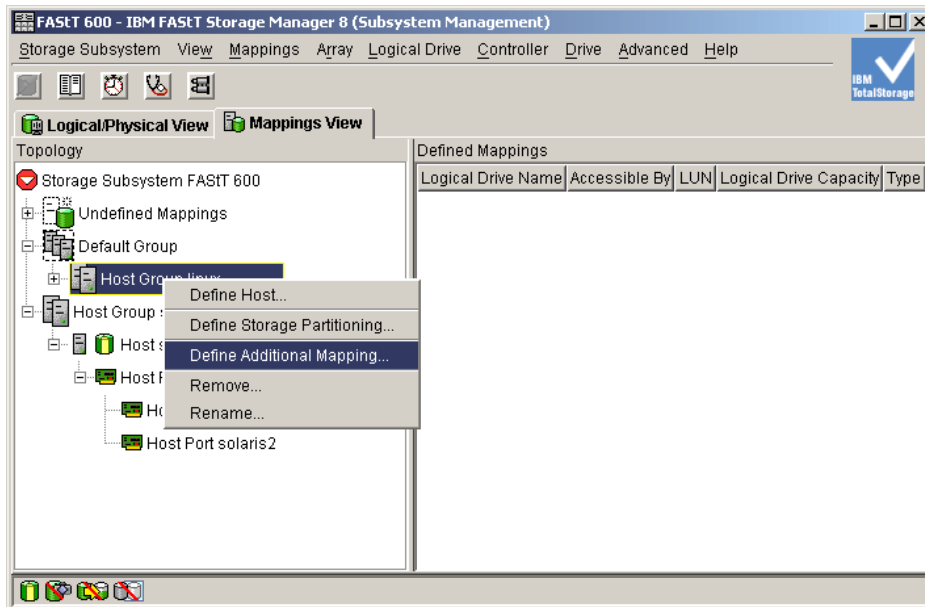


Figure 6-24 Define Additional Mapping for Linux

4. Right-click **Host Group Linux** and click **Define Additional Mapping**. You will see a window similar to Figure 6-25 on page 110.

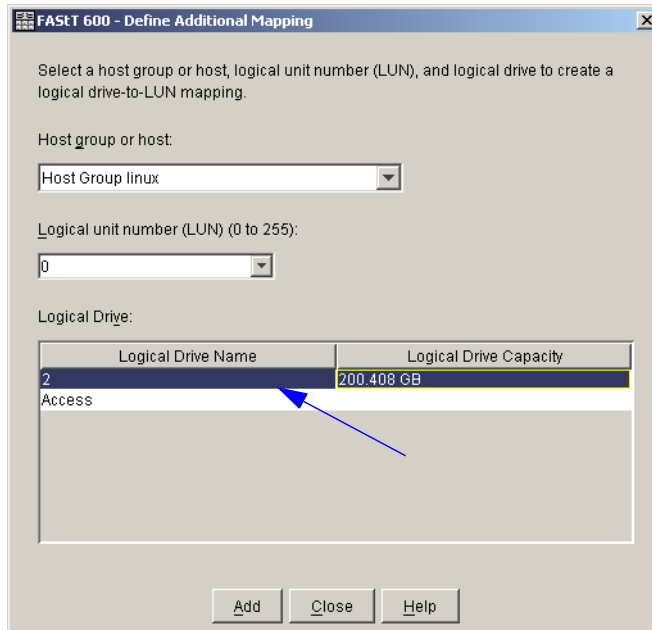


Figure 6-25 Define Additional Mapping window

5. Select logical drives, and click **Add**.
6. Click **Close**.

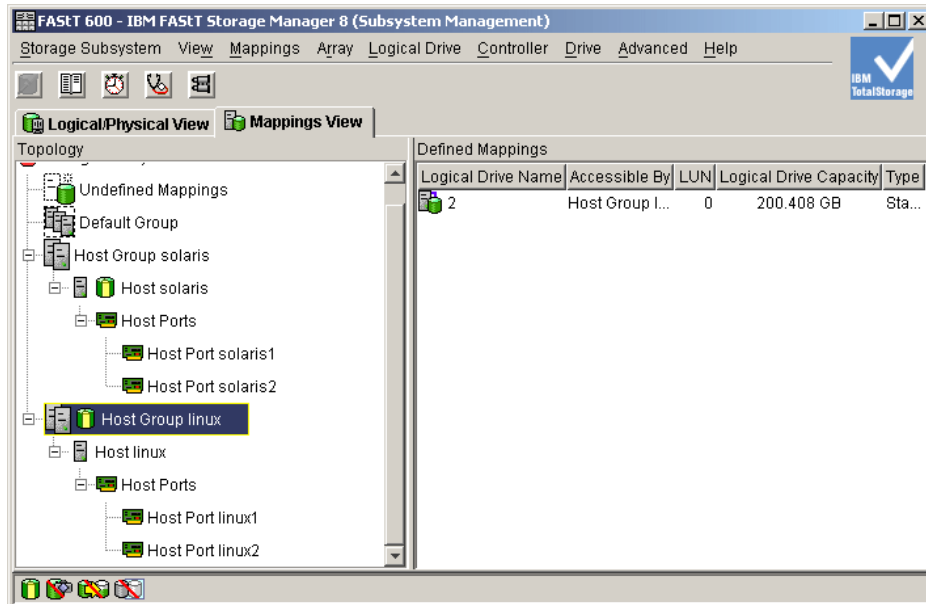


Figure 6-26 Mapping results

You must reboot the Linux machine for the LUN storage to be visible. After you reboot the Linux machine, import the *oracledg* disk group into the Volume Manager view. The file system */exp* is mounted at the same mount point on the Linux box. These operations must be performed by the root user (Volume Manager and File System operations). Run the commands highlighted in blue in Example 6-5 from the Linux side.

Example 6-5 DG import and file system mounting on the Linux server

```
Script started on Wed 15 Sep 2004 11:53:31 AM EDT
Linux # vxdisk list
DEVICE      TYPE          DISK          GROUP          STATUS
hda         auto:none    -             -              online invalid
sda         auto:cdsdisk -             -              online
sdc         auto         -             -              error
Linux # vx dg import oracledg
VxVM vx dg WARNING V-5-1-1328 Volume oracle0: Temporarily renumbered due to conflict
VxVM vx dg WARNING V-5-1-1328 Volume oracle1: Temporarily renumbered due to conflict
VxVM vx dg WARNING V-5-1-1328 Volume oracle2: Temporarily renumbered due to conflict
VxVM vx dg WARNING V-5-1-1328 Volume exp: Temporarily renumbered due to conflict
Linux # vxprint -ht
```

Disk group: oracledg

DG NAME	NCONFIG	NLOG	MINORS	GROUP-ID			
ST NAME	STATE	DM_CNT	SPARE_CNT		APPVOL_CNT		
DM NAME	DEVICE	TYPE	PRIVLEN	PUBLEN	STATE		
RV NAME	RLINK_CNT	KSTATE	STATE	PRIMARY	DATAVOLS	SRL	
RL NAME	RVG	KSTATE	STATE	REM_HOST	REM_DG	REM_RLNK	
CO NAME	CACHEVOL	KSTATE	STATE				
VT NAME	NVOLUME	KSTATE	STATE				
V NAME	RVG/VSET/CO	KSTATE	STATE	LENGTH	READPOL	PREFPLEX	UTYPE
PL NAME	VOLUME	KSTATE	STATE	LENGTH	LAYOUT	NCOL/WID	MODE
SD NAME	PLEX	DISK	DISKOFFS	LENGTH	[COL/]OFF	DEVICE	MODE
SV NAME	PLEX	VOLNAME	NVOLLAYR	LENGTH	[COL/]OFF	AM/NM	MODE
SC NAME	PLEX	CACHE	DISKOFFS	LENGTH	[COL/]OFF	DEVICE	MODE
DC NAME	PARENTVOL	LOGVOL					
SP NAME	SNAPVOL	DCO					

dg oracledg	default	default	9000	1095102646.12.sun1			
dm oracledg01	sda	auto	2048	420263680	-		
v exp	-	DISABLED	CLEAN	20971520	SELECT	-	fsgen
pl exp-01	exp	DISABLED	CLEAN	20971520	CONCAT	-	RW
sd oracledg01-04	exp-01	oracledg01	62914560	20971520	0	sda	ENA
v oracle0	-	DISABLED	CLEAN	20971520	SELECT	-	fsgen
pl oracle0-01	oracle0	DISABLED	CLEAN	20971520	CONCAT	-	RW
sd oracledg01-01	oracle0-01	oracledg01	0	20971520	0	sda	ENA
v oracle1	-	DISABLED	CLEAN	20971520	SELECT	-	fsgen
pl oracle1-01	oracle1	DISABLED	CLEAN	20971520	CONCAT	-	RW
sd oracledg01-02	oracle1-01	oracledg01	20971520	20971520	0	sda	ENA
v oracle2	-	DISABLED	CLEAN	20971520	SELECT	-	fsgen
pl oracle2-01	oracle2	DISABLED	CLEAN	20971520	CONCAT	-	RW
sd oracledg01-03	oracle2-01	oracledg01	41943040	20971520	0	sda	ENA
Linux # <code>vxvol -g oracledg startall</code> ←							
Linux # <code>vxprint -ht</code> ←							
Disk group: oracledg							

DG NAME	NCONFIG	NLOG	MINORS	GROUP-ID			
ST NAME	STATE	DM_CNT	SPARE_CNT		APPVOL_CNT		
DM NAME	DEVICE	TYPE	PRIVLEN	PUBLEN	STATE		
RV NAME	RLINK_CNT	KSTATE	STATE	PRIMARY	DATAVOLS	SRL	

RL NAME	RVG	KSTATE	STATE	REM_HOST	REM_DG	REM_RLNK	
CO NAME	CACHEVOL	KSTATE	STATE				
VT NAME	NVOLUME	KSTATE	STATE				
V NAME	RVG/VSET/CO	KSTATE	STATE	LENGTH	READPOL	PREFPLEX	UTYPE
PL NAME	VOLUME	KSTATE	STATE	LENGTH	LAYOUT	NCOL/WID	MODE
SD NAME	PLEX	DISK	DISKOFFS	LENGTH	[COL/]OFF	DEVICE	MODE
SV NAME	PLEX	VOLNAME	NVOLLAYR	LENGTH	[COL/]OFF	AM/NM	MODE
SC NAME	PLEX	CACHE	DISKOFFS	LENGTH	[COL/]OFF	DEVICE	MODE
DC NAME	PARENTVOL	LOGVOL					
SP NAME	SNAPVOL	DCO					

```

dg oracledg      default      default  9000      1095102646.12.sun1

dm oracledg01   sda          auto     2048      420263680 -

v  exp          -            ENABLED  ACTIVE    20971520 SELECT  -        fsgen

pl exp-01      exp          ENABLED  ACTIVE    20971520 CONCAT  -        RW
sd oracledg01-04 exp-01      oracledg01 62914560 20971520 0      sda     ENA

v  oracle0     -            ENABLED  ACTIVE    20971520 SELECT  -        fsgen

pl oracle0-01  oracle0     ENABLED  ACTIVE    20971520 CONCAT  -        RW
sd oracledg01-01 oracle0-01 oracledg01 0         20971520 0      sda     ENA

v  oracle1     -            ENABLED  ACTIVE    20971520 SELECT  -        fsgen

pl oracle1-01  oracle1     ENABLED  ACTIVE    20971520 CONCAT  -        RW
sd oracledg01-02 oracle1-01 oracledg01 20971520 20971520 0      sda     ENA

v  oracle2     -            ENABLED  ACTIVE    20971520 SELECT  -        fsgen

pl oracle2-01  oracle2     ENABLED  ACTIVE    20971520 CONCAT  -        RW
sd oracledg01-03 oracle2-01 oracledg01 41943040 20971520 0      sda     ENA

```

```

Linux # mkdir /exp
Linux # mount /dev/vx
Linux # mount -t vxfs /dev/vx/dsk/oracledg/exp /exp
Linux # head -10 /exp/exp-session
Script started on Wed 15 Sep 2004 11:30:26 AM EDT
Solaris # clear
Solaris # which exp
//ora17/oracle/product/8.1.7/bin/exp
Solaris # hostname
sun1
Solaris # uname -a
SunOS sun1 5.8 Generic_117350-06 sun4u sparc SUNW,Ultra-60
Solaris # set |grep ORACLE
ORACLE_BASE=/ora17/oracle

```

```
Linux # exit ←
exit
```

Script done on Wed 15 Sep 2004 11:56:46 AM EDT

---

## Step 5: Perform Oracle application import procedure on Linux

From a Linux command prompt, we ran the command **head -10** on the file `/exp/exp-session` on the Linux. Incidentally, this file was a session-capture on the Solaris and was created by Solaris.

After migrating and mounting on the Linux side, the same file is available as a text file. The next step is to ensure that the Oracle server side on the Linux is a proper environment for importing by running `@?/rdbms/admin/catalog` and `@?/rdbms/admin/catproc` after logging in through the sqlplus. Do this step and the actual step of running the import utility after logging into the Linux machine as the Oracle user. From a Linux shell prompt logging in as an Oracle user, run the following command (Example 6-6):

```
[oracle@bc3srv4 oracle]$imp system/manager FILE=/exp/ora8i.exp FULL=y
```

### Example 6-6 Query on Oracle on Linux

---

```
Script started on Wed Sep 15 13:18:35 2004
/u01/app/oracle
/u01/app/oracle/product/9.2.0.1.0
[oracle@bc3srv4 oracle]$imp system/manager FILE=/exp/ora8i.exp FULL=y ←
```

Import: Release 9.2.0.4.0 - Production on Wed Sep 15 13:19:19 2004

Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.

Connected to: Oracle9i Enterprise Edition Release 9.2.0.4.0 - Production  
With the Partitioning, OLAP and Oracle Data Mining options  
JServer Release 9.2.0.4.0 - Production

```
Export file created by EXPORT:V08.01.07 via conventional path
import done in US7ASCII character set and AL16UTF16 NCHAR character set
export server uses US7ASCII NCHAR character set (possible ncharset conversion)
. importing SYSTEM's objects into SYSTEM
IMP-00015: following statement failed because the object already exists:
"CREATE TABLE "DEF$_CALLDEST" ("ENQ_TID" VARCHAR2(22), "STEP_NO" NUMBER, "DB"
"LINK" VARCHAR2(128), "SCHEMA_NAME" VARCHAR2(30), "PACKAGE_NAME" VARCHAR2(30"
")) PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 LOGGING STORAGE(INITIAL 1"
"0240 NEXT 10240 MINEXTENTS 1 MAXEXTENTS 121 PCTINCREASE 50 FREELISTS 1 FREE"
```



```

"LIST GROUPS 1 BUFFER_POOL DEFAULT)
IMP-00015: following statement failed because the object already exists:
"CREATE TABLE "DEF$_DEFAULTDEST" ("DBLINK" VARCHAR2(128)) PCTFREE 10 PCTUSE
"D 40 INITRANS 1 MAXTRANS 255 LOGGING STORAGE(INITIAL 10240 NEXT 10240 MINEX
"TENTS 1 MAXEXTENTS 121 PCTINCREASE 50 FREELISTS 1 FREELIST_GROUPS 1 BUFFER_
"POOL DEFAULT)
IMP-00015: following statement failed because the object already exists:
"CREATE TABLE "DEF$_DESTINATION" ("DBLINK" VARCHAR2(128), "LAST_DELIVERED" N
"UMBER NOT NULL ENABLE, "LAST_ENQ_TID" VARCHAR2(22), "LAST_SEQ" NUMBER, "DIS
"ABLED" CHAR(1), "JOB" NUMBER, "LAST_TXN_COUNT" NUMBER, "LAST_ERROR_NUMBER"
"NUMBER, "LAST_ERROR_MESSAGE" VARCHAR2(2000)) PCTFREE 10 PCTUSED 40 INITRAN
"S 1 MAXTRANS 255 LOGGING STORAGE(INITIAL 10240 NEXT 10240 MINEXTENTS 1 MAXE
"XTENTS 121 PCTINCREASE 50 FREELISTS 1 FREELIST_GROUPS 1 BUFFER_POOL DEFAULT
")
IMP-00015: following statement failed because the object already exists:
"CREATE TABLE "DEF$_ERROR" ("ENQ_TID" VARCHAR2(22), "ORIGIN_TRAN_DB" VARCHAR
"2(128), "ORIGIN_ENQ_TID" VARCHAR2(22), "DESTINATION" VARCHAR2(128), "STEP_N
"O" NUMBER, "RECEIVER" NUMBER, "ENQ_TIME" DATE, "ERROR_NUMBER" NUMBER, "ERRO
"R_MSG" VARCHAR2(2000)) PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 LOGGI
"NG STORAGE(INITIAL 10240 NEXT 10240 MINEXTENTS 1 MAXEXTENTS 121 PCTINCREASE
" 50 FREELISTS 1 FREELIST_GROUPS 1 BUFFER_POOL DEFAULT)
IMP-00015: following statement failed because the object already exists:
"CREATE TABLE "DEF$_LOB" ("ID" RAW(16), "ENQ_TID" VARCHAR2(22), "BLOB_COL" B
"LOB, "CLOB_COL" CLOB, "NCLOB_COL" NCLOB) PCTFREE 10 PCTUSED 40 INITRANS 1
"MAXTRANS 255 LOGGING STORAGE(INITIAL 10240 NEXT 10240 MINEXTENTS 1 MAXEXTEN
"TS 121 PCTINCREASE 50 FREELISTS 1 FREELIST_GROUPS 1 BUFFER_POOL DEFAULT) TA
"BLESPACE "SYSTEM" LOB ("BLOB_COL") STORE AS (TABLESPACE "SYSTEM" ENABLE ST
"ORAGE IN ROW CHUNK 2048 PCTVERSION 0 NOCACHE STORAGE(INITIAL 10240 NEXT 10
"240 MINEXTENTS 1 MAXEXTENTS 121 PCTINCREASE 50 FREELISTS 1 FREELIST_GROUPS
" 1 BUFFER_POOL DEFAULT)) LOB ("CLOB_COL") STORE AS (TABLESPACE "SYSTEM" ENA
"BLE STORAGE IN ROW CHUNK 2048 PCTVERSION 0 NOCACHE STORAGE(INITIAL 10240 N
"EXT 10240 MINEXTENTS 1 MAXEXTENTS 121 PCTINCREASE 50 FREELISTS 1 FREELIST G
"ROUPS 1 BUFFER_POOL DEFAULT)) LOB ("NCLOB_COL") STORE AS (TABLESPACE "SYST
"EM" ENABLE STORAGE IN ROW CHUNK 2048 PCTVERSION 0 NOCACHE STORAGE(INITIAL
" 10240 NEXT 10240 MINEXTENTS 1 MAXEXTENTS 121 PCTINCREASE 50 FREELISTS 1 FRE
"ELIST_GROUPS 1 BUFFER_POOL DEFAULT))"
IMP-00015: following statement failed because the object already exists:
"CREATE TABLE "DEF$_ORIGIN" ("ORIGIN_DB" VARCHAR2(128), "ORIGIN_DBLINK" VARC
"HAR2(128), "INUSR" NUMBER, "CSCN" NUMBER, "ENQ_TID" VARCHAR2(22), "RECO_SEQ
" _NO" NUMBER) PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 LOGGING STORAGE
"(INITIAL 10240 NEXT 10240 MINEXTENTS 1 MAXEXTENTS 121 PCTINCREASE 50 FREELI
"STS 1 FREELIST_GROUPS 1 BUFFER_POOL DEFAULT)
IMP-00015: following statement failed because the object already exists:
"CREATE TABLE "DEF$_PROPAGATOR" ("USERID" NUMBER, "USERNAME" VARCHAR2(30) NO
" T NULL ENABLE, "CREATED" DATE NOT NULL ENABLE) PCTFREE 10 PCTUSED 40 INITR
"ANS 1 MAXTRANS 255 LOGGING STORAGE(INITIAL 10240 NEXT 10240 MINEXTENTS 1 MA
"XEXTENTS 121 PCTINCREASE 50 FREELISTS 1 FREELIST_GROUPS 1 BUFFER_POOL DEFAU
"LT)
IMP-00015: following statement failed because the object already exists:

```

```

"CREATE TABLE "DEF$_PUSHED_TRANSACTIONS" ("SOURCE_SITE_ID" NUMBER, "LAST_TRA
"N_ID" NUMBER, "DISABLED" VARCHAR2(1), "SOURCE_SITE" VARCHAR2(128)) PCTFREE"
" 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 LOGGING STORAGE(INITIAL 10240 NEXT 1"
"0240 MINEXTENTS 1 MAXEXTENTS 121 PCTINCREASE 50 FREELISTS 1 FREELIST GROUPS"
" 1 BUFFER_POOL DEFAULT)
"
IMP-00015: following statement failed because the object already exists:
"CREATE TABLE "DEF$_TEMP$LOB" ("TEMP$BLOB" BLOB, "TEMP$CLOB" CLOB, "TEMP$NCL
"OB" NCLOB) PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOLOGGING STORAGE"
"(INITIAL 10240 NEXT 10240 MINEXTENTS 1 MAXEXTENTS 121 PCTINCREASE 50 FREELI
"STS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT) TABLESPACE "SYSTEM" LOB ("TEMP
"$BLOB") STORE AS (TABLESPACE "SYSTEM" ENABLE STORAGE IN ROW CHUNK 2048 PCT"
"VERSION 0 NOCACHE NOLOGGING STORAGE(INITIAL 10240 NEXT 10240 MINEXTENTS 1 "
"MAXEXTENTS 121 PCTINCREASE 50 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEF
"AULT)) LOB ("TEMP$CLOB") STORE AS (TABLESPACE "SYSTEM" ENABLE STORAGE IN R"
"OW CHUNK 2048 PCTVERSION 0 NOCACHE NOLOGGING STORAGE(INITIAL 10240 NEXT 10"
"240 MINEXTENTS 1 MAXEXTENTS 121 PCTINCREASE 50 FREELISTS 1 FREELIST GROUPS "
"1 BUFFER_POOL DEFAULT)) LOB ("TEMP$NCLOB") STORE AS (TABLESPACE "SYSTEM" E"
"NABLE STORAGE IN ROW CHUNK 2048 PCTVERSION 0 NOCACHE NOLOGGING STORAGE(INI"
"TIAL 10240 NEXT 10240 MINEXTENTS 1 MAXEXTENTS 121 PCTINCREASE 50 FREELISTS "
"1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT))"
. . importing table          "EMP"          16384 rows imported
. . importing table          "SQLPLUS_PRODUCT_PROFILE"          0 rows imported
IMP-00015: following statement failed because the object already exists:
"CREATE SYNONYM "CATALOG" FOR "SYS"."CATALOG"
IMP-00015: following statement failed because the object already exists:
"CREATE SYNONYM "SYSCATALOG" FOR "SYS"."SYSCATALOG"
IMP-00015: following statement failed because the object already exists:
"CREATE SYNONYM "TAB" FOR "SYS"."TAB"
IMP-00015: following statement failed because the object already exists:
"CREATE SYNONYM "COL" FOR "SYS"."COL"
IMP-00015: following statement failed because the object already exists:
"CREATE SYNONYM "TABQUOTAS" FOR "SYS"."TABQUOTAS"
IMP-00015: following statement failed because the object already exists:
"CREATE SYNONYM "PUBLICSYN" FOR "SYS"."PUBLICSYN"
IMP-00015: following statement failed because the object already exists:
"CREATE SYNONYM "SYSFILES" FOR "SYS"."SYSFILES"
Import terminated successfully with warnings.
exit ←

```

Script done on Wed Sep 15 13:24:23 2004

---

## Step 6: Verify the migration

The Oracle database migration has taken place. The warnings created above were due to a request of creating files for the system and manager related table objects. We are particularly interested in the table EMP in this migration case, and it can be noted that the table for EMP has been successfully imported. This brings us to the verification stage of the two database, the source on Solaris and

the target on the Linux machine. The same query is run on both the instances. The response of the two servers is identical to the query ( Example 6-7 and Example 6-8).

*Example 6-7 Solaris side query for the table EMP and its response*

---

```
Script started on Wed 15 Sep 2004 04:20:19 PM EDT
Solaris # svrmgr1 ←

Oracle Server Manager Release 3.1.7.0.0 - Production

Copyright (c) 1997, 1999, Oracle Corporation. All Rights Reserved.

Oracle8i Enterprise Edition Release 8.1.7.0.0 - 64bit Production
With the Partitioning option
JServer Release 8.1.7.0.0 - 64bit Production

SVRMGR> connect system/manager ←
Connected.
SVRMGR> select count(*) from emp; ←
COUNT(*)
-----
      16384
1 row selected.
SVRMGR> exit ←
Server Manager complete.
Solaris # exit ←
exit

script done on Wed 15 Sep 2004 04:21:16 PM EDT
```

---

*Example 6-8 Linux side query for the table EMP and its response*

---

```
Script started on Wed Sep 15 15:57:35 2004
/u01/app/oracle
/u01/app/oracle/product/9.2.0.1.0
[oracle@bc3srv4 oracle]$ sqlplus "/ as sysdba" ←

SQL*Plus: Release 9.2.0.4.0 - Production on Wed Sep 15 15:57:46 2004

Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.

Connected to:
```

Oracle9i Enterprise Edition Release 9.2.0.4.0 - Production  
With the Partitioning, OLAP and Oracle Data Mining options  
JServer Release 9.2.0.4.0 - Production

```
SQL> connect system/manager ←  
Connected.  
SQL> select count(*) from emp; ←
```

```
  COUNT(*)  
-----  
      16384
```

```
SQL> exit ←  
Disconnected from Oracle9i Enterprise Edition Release 9.2.0.4.0 - Production  
With the Partitioning, OLAP and Oracle Data Mining options  
JServer Release 9.2.0.4.0 - Production  
[oracle@bc3srv4 oracle]$ exit ←  
exit
```

Script done on Wed Sep 15 15:58:09 2004

---

To conclude, the demonstration of the Oracle application from Solaris to Linux side was archived with the help of the VERITAS Portable Data Container technology and a step-by-step process of the same has been documented. Although the problem size for demonstration was small, it does depict a real-world migration scenario. With proper care and knowledge of the various restrictions and capabilities of the various products used, it is possible to do a migration of Oracle on Solaris to Oracle on Linux.

## 6.7 DB2 application migration

This section discusses the steps necessary for a successful DB2 application migration.

To migrate DB2 from UNIX to Linux, perform the following steps:

1. Install DB2 8.1 on Solaris 9.
2. Install DB2 8.1 on Linux Red Hat AS 3.0.
3. Create a sample database on source side DB2.
4. Export the database objects and tables using the db2look and db2move utilities of DB2 8.1
5. Perform the Portable Data Containers conversion step on the Solaris side

6. Unmount of the VERITAS File System and deport the disk group on Solaris side.
7. Import the disk group, start the volumes, and mount the VERITAS File System on the Linux side.
8. Create a database on the Linux side.
9. Run the SQL created using db2look.
10. Perform an import procedure on Linux.

A successful migration happens if same the DB2 servers on the source and target platform respond with an identical response.

## **Create the diskgroup on which the database will reside on Solaris**

Example 6-9 provides the necessary steps to create a diskgroup on the Solaris server. From a Solaris command prompt, execute the commands in the example.

### *Example 6-9 Diskgroup creation*

---

```
As root:
Solaris $ uname -a
SunOS sun1 5.9 Generic_112233-10 sun4u sparc SUNW,Sun-Fire-V240

Solaris # vxdg init oridg disk1=c1t1d0s1 disk2=c1t2d0s2
Note: both the disks are created as cdsdisks

Solaris # vxassist -g oridg make db2vol 1g alloc=disk1
Solaris # mkfs -F vxfs /dev/vx/rdisk/oridg/db2vol
Solaris # mkdir /db2fs
Solaris # mount -F vxfs /dev/vx/dsk/oridg/db2vol /db2fs
Solaris # chown -R inst1 /db2fs
```

---

## **Create another volume and file system that will be used to store the exported data**

Create a volume to export the data. Complete the instructions in Example 6-10.

### *Example 6-10 Volume creation*

---

```
Solaris # vxassist -g oridg make migvol 1g alloc=disk2
Solaris # mkfs -F vxfs /dev/vx/rdisk/oridg/migvol
Solaris # mkdir /migfs
Solaris # mount -F vxfs /dev/vx/dsk/oridg/migvol /migfs
Solaris # chown -R inst1 /migfs
Solaris # vxprint -g oridg
```

TY NAME	ASSOC	KSTATE	LENGTH	PLOFFS	STATE	TUTILO	PUTILO
dg oridg	oridg	-	-	-	-	-	-
dm disk1	c1t1d0s1	-	35137584	-	-	-	-
dm disk2	c1t2d0s2	-	35137584	-	-	-	-
v db2vol	fsgen	ENABLED	2097152	-	ACTIVE	-	-
p1 db2vol-01	db2vol	ENABLED	2097152	-	ACTIVE	-	-
sd disk1-01	db2vol-01	ENABLED	2097152	0	-	-	-
v migvol	fsgen	ENABLED	2097152	-	ACTIVE	-	-
p1 migvol-01	migvol	ENABLED	2097152	-	ACTIVE	-	-
sd disk2-01	migvol-01	ENABLED	2097152	0	-	-	-

The DB administrator should ensure that the instance *inst1* has been created and then proceed with the next step.

## Create the sample DB2 database on /db2fs

Follow the steps in Example 6-11 to create a sample database for the DB2 migration.

### Example 6-11 Database creation

---

```
Solaris $ db2 create db empdb on /db2fs
Solaris $ db2 connect to empdb
Solaris $ db2 "create tablespace emptbs managed by database using (file
'/db2fs/emptbs' 5000)"
Solaris $ db2 "create table emp (id number, name char(1024), addr
varchar(2048))
Solaris $ db2 insert into emp values (1, 'first name last name', 'One Universal
Dr.')
```

```
Solaris $ db2 insert into emp select * from emp
Solaris $ db2 insert into emp select * from emp
Solaris $ db2 insert into emp select * from emp
Solaris $ db2 insert into emp select * from emp
Solaris $ db2 insert into emp select * from emp
Solaris $ db2 "select count(*) from emp"
1
-----
          32
1 record(s) selected.
```

---

## Export the database objects and table data on DB2 8.1 running on Solaris

Complete the instructions in Example 6-12 to export the database from the Solaris source and to prepare for importing the Linux server.

### *Example 6-12 Export database from Solaris*

---

```
Solaris $ db2level
DB21085I Instance "inst1" uses "32" bits and DB2 code release "SQL08010" with
level identifier "01010106".
Informational tokens are "DB2 v8.1.0.0", "s021023", "", and FixPak "0".
Product is installed at "/opt/IBM/db2/V8.1".
```

```
Solaris $ echo $DB2INSTANCE
inst1
```

```
Solaris $ echo $DB2DATABASE
empdb
```

```
Solaris $ db2look -d empdb -e -a -o /migfs/emplook.sql
Solaris $ db2move empdb export
```

```
Solaris # umount /migvol
Solaris # fscdsconv -y -f /tmp/db2fs.conv /dev/vx/rdisk/oridg/migvol
UX:vxfs fscdsconv: INFO: V-3-21842: Do you wish to commit to conversion? (ynq)
y
```

```
Solaris # vxdg split oridg db2dg migvol
Solaris # vxdg deport db2dg
```

Importing the diskgroup

As root

```
Linux # uname -a
```

```
Linux Linux1 2.4.21-4.ELsmp #1 SMP Fri Oct 3 17:52:56 EDT 2003 i686 i686 i386
GN
```

U/Linux

```
Linux #vxdg import db2dg
```

```
Linux# mkdir /db2fs
```

```
Linux# vxvol -g db2dg startall
```

```
Linux# mount -t vxfs /dev/vx/dsk/db2dg/migvol /db2fs
```

```
Linux# chown -R inst1 /db2fs
```

```
Linux# vxprint -g db2dg
```

TY	NAME	ASSOC	KSTATE	LENGTH	PLOFFS	STATE	TUTILO	PUTILO
dg	suzpdcg	suzpdcg	-	-	-	-	-	-
dm	disk2	sd1	-	35137584	-	-	-	-
v	migvol	fsgen	ENABLED	2097152	-	ACTIVE	-	-

```
p1 migvol-01 migvol    ENABLED 2097152 -      ACTIVE -      -
sd disk2-01    migvol-01 ENABLED 2097152 0      -      -      -
```

---

## Ensure instance has been created

The DB administrator should ensure that the instance *inst1* has been created on the Linux server. Execute the command in Example 6-13.

### *Example 6-13 Execute db2level*

---

```
Linux $ db2level
DB21085I Instance "inst1" uses "32" bits and DB2 code release "SQL08020" with
level identifier "03010106".
Informational tokens are "DB2 v8.1.0.64", "s040415", "MI00060", and FixPak "7".
Product is installed at "/opt/IBM/db2/V8.1".
```

---

## Create the database and import

Create the database and complete the importing of the data.

### *Example 6-14 Database creation and import commands*

---

```
Linux $ db2 create database empdb on /db2fs
Linux $ db2 -tvf emplook.sql
Linux $ cd /db2fs
Linux $ db2move empdb import -l .
Linux $ db2 connect to empdb
Linux $ db2 "select count(*) from emp"
1
-----
          32
1 record(s) selected.
```

---





# A

## Manual Pages

This appendix includes the manual pages for the **fsadm\_vxfs** and **fscdsconv** commands.

### VERITAS File System, fsadm\_vxfs

#### Name

**fsadm\_vxfs** resizes or reorganizes (defragment) a VERITAS File System

#### Synopsis

```
fsadm [-t vxfs] [-d] [-D] [-e] [-E] [-s] [-v] [-a days] [-k ckpt_name] [-l  
largesize] [-p passes] [-r rawdev] [-t time] mount_point
```

```
fsadm [-t vxfs] [-b newsize] [-r rawdev] mount_point
```

```
fsadm [-t vxfs] [-o specific_options] mount_point | special
```

```
fsadm [-t vxfs] [-d] [-D] [-e] [-E] -f filename | -
```

```
fsadm [-t vxfs] [-L] [-o specific_options] mount_point
```

#### Availability

VRTSvxfs

## Description

The **fsadm** command performs online administration functions on a VERITAS File System, Storage Checkpoints, or individual files and directories. The **fsadm** command supports file system resizing, extent reorganization, directory reorganization, and querying or changing the `largefiles` flag. The **fsadm** command operates on file systems mounted for read and write access; however, the `-o` option can also operate on a special device containing a clean, unmounted file system. Only a privileged user can change the `largefiles` flag on a mounted file system, or resize or reorganize a file system. You can invoke only one instance of **fsadm** per file system at a time.

If `mount_point` is a Storage Checkpoint, **fsadm** performs the specified operation on the entire file system, including all of its Storage Checkpoints.

## Cluster File System Issues

For a cluster mounted file system, you must run **fsadm** from the primary node of the cluster, and the primary must be mounted with read and write access. **fsadm** fails if run from one of the secondary mounts. You can use the **fsclustadm showprimary mount\_point** command to determine which cluster node is the primary before you run **fsadm**.

## Options

If no options are specified, **fsadm** prints the current `largefiles` flag setting, then exits. The `-b`, `-o largefiles`, and `-o nolargefiles` options cannot be specified if any other options are given. If both `-e` and `-d` are specified, **fsadm** first completes the directory reorganization, then does the extent reorganization.

- |                   |  |
|-------------------|--|
| <b>-a days</b>    | Considers files not accessed within the specified number of days as <i>aged</i> files. The default is 14 days. The <code>-d</code> option moves aged files to the end of the directory.                                |
| <b>-b newsize</b> | Resizes the file system to newsize sectors. <code>-b</code> cannot be used with the <code>-dDeEo</code> options.   |
| <b>-d</b>         | Reorganizes directories. Directory entries are reordered to place subdirectory entries first, then all other entries in decreasing order of time of last access. The directory is also compacted to remove free space. |
| <b>-D</b>         | Reports on directory fragmentation. If specified with the <code>-d</code> option, the fragmentation report is produced both before and after the directory reorganization.   |
| <b>-e</b>         | Extent reorganization. Minimizes file system fragmentation. Files are reorganized to have the minimum number of extents.   |

<b>-E</b>	Reports on extent fragmentation. If specified with the <b>-e</b> option, the fragmentation report is produced both before and after the extent reorganization.
<b>-f filename   -</b>	Performs and reports extent reorganization or directory reorganization on an individual file or directory. For example, when used with the <b>-d</b> option, a reorganization is performed on the specified directory. When used with the <b>-e</b> option, an extent reorganization is performed on a specified file. If you specify <b>-</b> (dash), the command reads input from the standard input device instead of using a file name or directory name.
<b>-t vxfs</b>	Specifies the VERITAS File System type.
<b>-k ckpt_name</b>	Reorganizes or reports directories or extent fragmentation on the specified Storage Checkpoint.
<b>-l largesize</b>	Large extent size in file system blocks. Indicates the size of extents to be considered as large extents, that is, extents that are immovable while performing an extent defragmentation. The value must be between 8 and 2048 blocks. The default is 64 blocks.
<b>-L</b>	Queries the current size of the intent log as well as the volume on which it resides. On single-volume file systems, logvol will be empty.
<b>-o specific_options</b>	Specifies options specific to the VERITAS File System.
<b>largefiles</b>	Sets the largefiles flag for the file system. When this flag is set, large files (greater than two gigabytes) can be created on the file system.
<b>logsize=size</b>	Allocates the intent log size in file system blocks for the specified file system. The recommended value for size is greater than 256K. The maximum is 256 MB. Size can also be specified with a suffix to indicate a unit of measure other than file system blocks. Append <b>k</b> or <b>K</b> to indicate the value is in kilobytes, <b>m</b> or <b>M</b> to indicate megabytes, or <b>g</b> or <b>G</b> to indicate gigabytes. An appended letter can be separated from the number by a space. In that case, enclose the letter and number in a set of quotes (for example, "512 k"). Note: System performance can improve by increasing the size of the intent log because it reduces the frequency at which the log wraps. Increasing the intent log size can also increase the time required for a log replay.

<b>logvol=volumentname</b>	<p>Directs the intent log size allocation specified by logsize to the specified intent logging volume. volumename must exist and belong to the same volume set as the file system. This modifies the file system log allocation policy to contain the specified device. The old log space is freed after the command completes.</p> <p>If logvol is specified and logsize is not specified, the log is moved to the new volume while retaining the existing size.</p>
<b>no largefiles</b>	<p>Clears the largefiles flag for the file system. When the largefiles flag is not set, large files cannot be created on the file system. Any attempt to clear the flag fails if a large file exists on the file system.</p> <p>You can set or clear the flag on a mounted file system by specifying mount_point, or on an unmounted file system on the device special.</p> <p>When invoked without arguments, fsadm prints the current state of the largefiles flag.</p> <p>Note: Be careful when implementing large file system capability. System administration utilities such as backup may not operate correctly if they are not large file aware.</p> <p>The <b>-o</b> option cannot be used with the <b>-bdDeEfk</b> options.</p>
<b>-p</b>	<p>passes Maximum number of passes to run. The default is five passes.</p> <p>Reorganizations are processed until reorganization is complete, or until the specified number of passes are run.</p>
<b>-r</b>	<p>rawdev Path name of the raw device to read to determine file layout and fragmentation. This option can be used when fsadm cannot determine the raw device.</p>
<b>-s</b>	<p>Prints a summary of activity at the end of each pass.</p>
<b>-t time</b>	<p>Maximum time to run. Reorganizations are processed until reorganization is complete, or until the time limit expires.</p> <p>Time is specified in seconds.</p>
<b>-v</b>	<p>Specifies verbose mode. Reports reorganization activity.</p>
<b>Largefiles Flag</b>	<p>Large files (files larger than 2 gigabytes) can be created if the file system has the largefiles flag set. This flag is set by default when the file system is created.</p>

The **-o largefiles** and **-o nolargefiles** options change the largefiles flag, allowing or disallowing large files (files larger than 2 GB) in the file system. Clearing the flag via the **-o nolargefiles** option succeeds only if the flag is set and there are no large files present on the file system. See the **mkfs\_vxfs(1M)** and **mount\_vxfs(1M)** manual pages for information about creating and mounting file systems with large files.

The **-o largefiles** and **-o nolargefiles** options are the only fsadm options that can be used on an unmounted file system. An unmounted file system can be specified by invoking fsadm with a special device rather than a mount point. If you specify an unmounted file system, that file system must be clean.

Changing the largefiles flag may require changes to /etc/fstab. For example, if fsadm is used to set the largefiles flag, but nolargefiles is specified as a mount option in /etc/fstab, the file system is not mountable.

## Defragmentation

For optimal performance, the kernel-extent allocator must be able to find large extents when it wants them. To maintain file system performance, run **fsadm** periodically against all VERITAS File Systems to reduce fragmentation. The frequency depends on file system usage and activity patterns, and the importance of performance; typically between once a day and once a month against each file system. The **-v** option can be used to examine the amount of work performed by fsadm. You can adjust the frequency of reorganization based on the rate of file system fragmentation.

There are two options that are available to control the amount of work done by fsadm. The **-t** option specifies a maximum length of time to run. The **-p** option specifies a maximum number of passes to run. If both are specified, **fsadm** exits if either of the terminating conditions is reached. By default, **fsadm** runs five passes. If both the **-e** and **-d** options are specified, **fsadm** runs all the directory reorganization passes before any extent reorganization passes.

**fsadm** uses the file **.fsadm** in the lost+found directory as a lock file. When **fsadm** is invoked, it opens the file **lost+found/.fsadm** in the root of the file system specified by **mount\_point**. If **.fsadm** does not exist, it is created. The **fcntl** system call (see the **fcntl(2)** manual page) obtains a write lock on the file. If the write lock fails, **fsadm** assumes that another instance of **fsadm** is running and fails. **fsadm** reports the process ID of the process holding the write lock on the **.fsadm** file.

## File System Resizing

If the **-b** option is specified, fsadm resizes the file system whose mount point is **mount\_point**. If **newsize** is larger than the current size of the file system, the file system is expanded to **newsize** sectors. Similarly, if **newsize** is smaller than the current size of the file system, fsadm shrinks the file system to **newsize** sectors.

If there are file system resources in use in the sectors being removed, `fsadm` relocates those resources to sectors staying within the resized file system. The time needed for relocation depends on the number of blocks being moved.

In some circumstances, the `fsadm` command cannot resize a 100% full file system due to lack of space for updating structural information. Check VxFS file systems on a regular basis; increase their size if they approach 100% capacity. This problem can also occur if the file system is very busy. Free up space or reduce activity on the file system and try the resize again.

The `fsadm` command cannot resize multi-volume file systems. See the `fsvoladm(1M)` manual page for information about resizing multi-volume file systems.

## Reporting On Directory Fragmentation

As files are allocated and deallocated, directories tend to grow and become sparse. In general, a directory is as large as the largest number of files it ever contained, even if some files were subsequently removed.

To obtain a directory fragmentation report, use the command syntax:

```
fsadm -D [-r rawdev] mount_point
```

The following is some example output from the `fsadm -D` command:

```
# fsadm -t vxfs -D /var
```

### *Directory Fragmentation Report*

Dirs Searched	Total Blocks	Immed Dirs	Immeds to Add	Dirs to Reduce	Blocks to Reduce
total 486	99	388	6	6	6

The Dirs Searched column contains the total number of directories. A directory is associated with the extent-allocation unit containing the extent in which the directory's inode is located. The Total Blocks column contains the total number of blocks used by directory extents.

The Immed Dirs column contains the number of directories that are immediate, meaning that the directory data is in the inode itself, as opposed to being in an extent. Immediate directories save space and speed up path name resolution. The Immed to Add column contains the number of directories that currently have a data extent, but that could be reduced in size and contained entirely in the inode.

The Dirs to Reduce column contains the number of directories for which one or more blocks could be freed if the entries in the directory are compressed to make the free space in the directory contiguous.

Because directory entries vary in length, it is possible that some large directories may contain a block or more of total free space, but with the entries arranged in such a way that the space cannot be made contiguous. As a result, it is possible to have a non-zero Dirs to Reduce calculation immediately after running a directory reorganization. The **-v** (verbose) option of directory reorganization reports occurrences of failure to compress free space.

The Blocks to Reduce column contains the number of blocks that could be freed if the entries in the directory are compressed.

## Measuring Directory Fragmentation

If the totals in the columns labeled Dirs to Reduce are substantial, a directory reorganization can improve the performance of path name resolution. The directories that fragment tend to be the directories with the most activity. A small number of fragmented directories can account for a large percentage of name lookups in the file system.

## Directory Reorganization

If the **-d** option is specified, **fsadm** reorganizes the directories on the file system whose mount point is `mount_point`. Directories are reorganized in two ways: compression and sorting.

For compression, **fsadm** moves valid entries to the front of the directory and groups the free space at the end of the directory. If there are no entries in the last block of the directory, the block is released and the directory size is reduced.

If the total space used by all directory entries is small enough, **fsadm** puts the directory in the inode immediate data area. **fsadm** also sorts directory entries to improve path name lookup performance. Entries are sorted based on the last access time of the entry.

The **-a** option specifies a time interval; 14 days is the default if **-a** is not specified. The time interval is broken up into 128 buckets, and all times within the same buckets are considered equal. All access times older than the time interval are considered equal, and those entries are placed last. Subdirectory entries are placed at the front of the directory and symbolic links are placed after subdirectories, followed by the most-recently-accessed files.

The command syntax for reorganizing directories in a file system is:

```
fsadm -d [-D][-s][-v][-a days][-p passes][-t time][-r rawdev] mount_point
```

The following example shows the output of the **fsadm -d -D** command:

```
# fsadm -t vxfs -d -D -s /opt
```

### ***Directory Fragmentation Report***

Dirs	Total	Immed	Immeds	Dirs to	Blocks
Searched	Blocks	Dirs	to Add	Reduce	to Reduce
total	34663	8800	26655	2569	2716 2836

### ***Directory Reorganization Statistics (pass 1 of 2)***

Dirs	Dirs	Total	Failed	Blocks	Blocks	Immeds
Searched	Changed	Ioctl's	Ioctl's	Reduced	Changed	Added
fset 999	8008	3121	5017	0	3037	4428 2569
total	8008	3121	5017	0	3037	4428 2569

### ***Directory Reorganization Statistics (pass 2 of 2)***

Dirs	Dirs	Total	Failed	Blocks	Blocks	Immeds
Searched	Changed	Ioctl's	Ioctl's	Reduced	Changed	Added
fset 999	5439	552	2448	0	708	4188 0
total	5439	552	2448	0	708	4188 0

### ***Directory Fragmentation Report***

Dirs	Total	Immed	Immeds	Dirs to	Blocks
Searched	Blocks	Dirs	to Add	Reduce	to Reduce
total	34663	6231	29224	0	147 267

The Dirs Searched column contains the number of directories searched. Only directories with data extents are reorganized. Immediate directories are skipped. The Dirs Changed column contains the number of directories for which a change was made.

The Total ioctl's column contains the total number of VX\_DIRSORT ioctl's performed. Reorganization of directory extents is performed using this ioctl.

The Failed ioctl's column contains the number of requests that failed. The reason for failure is usually that the directory being reorganized is active. A few failures are typical in most file systems. If the `-v` option is specified, all ioctl calls and status returns are recorded.

The Blocks Reduced column contains the total number of directory blocks freed by compressing entries. The Blocks Changed column contains the total number of directory blocks updated while sorting and compressing entries.

The Immeds Added column contains the total number of directories with data extents that were compressed into immediate directories.

### **Reporting on Extent Fragmentation**

As files are created and removed, over time the free extent map for an allocation unit changes from having one large free area to having many smaller free areas. This process is known as fragmentation. Also, when files increase in size



(particularly when growth occurs in small increments), small files can be allocated in multiple extents. In the best case, each file that is not sparse has exactly one extent (containing the entire file), and the free-extent map is one continuous range of free blocks.

## Determining Fragmentation

To determine whether a file system is fragmented, the free extents for that file system must be examined. If a large number of small extents are free, there is fragmentation. If more than half of the amount of free space is used by small extents (smaller than 64 blocks), or there is less than 5 percent of total file system space available in large extents, there is serious fragmentation.

## Running The Extent-fragmentation Report

The extent-fragmentation report provides detailed information about the degree of fragmentation in a given file system.

The command syntax for an extent-fragmentation report is:

```
fsadm -E [-l largesize] [-r rawdev] mount_point
```

The extent reorganization facility considers certain extents to be immovable, that is, if the file already contains large extents, reallocating and consolidating these extents does not improve performance.

The `-l` option controls when `fsadm` considers an extent immovable. By default, `largesize` is 64 blocks. For the extent-fragmentation report, the value for `largesize` affects which extents are reported as being immovable extents.

The following is an example of the output generated by the `fsadm -E` command:

```
# fsadm -t vxfs -E /home
```

### ***Extent Fragmentation Report***

```
Total      Average      Average      Total
Files      File Blks    # Extents    Free Blks
939         11           2            245280
blocks used for indirects: 0
% Free blocks in extents smaller than 64 blks: 8.35
% Free blocks in extents smaller than 8 blks: 4.16
% blks allocated to extents 64 blks or larger: 45.81
Free Extents By Size
1: 356      2: 292      4: 271      8: 181
16: 23      32: 7       64: 3       128: 1
256: 1      512: 0      1024: 1     2048: 1
4096: 2     8192: 2     16384: 1    32768: 2
```

The numbers in the Total Files column indicate the total number of files that have data extents. The Average File Blks column contains the average number of blocks belonging to all files. The Average # Extents column contains the average number of extents used by files in the file system. The Total Free Blks column contains the total number of free blocks in the file system. The total number of blocks used for indirect address extent are reported as *blocks used for indirects*.

The general shape of free extent map is also reported. There are two percentages reported, percentage of free extents smaller than 64 blocks and percentage of free extents smaller than 8 blocks. These numbers are typically near zero on an unfragmented file system.

Another metric reported is the percentage of blocks that are part of extents 64 blocks or larger. Files with a single small extent are not included in this calculation. This number is typically large on file systems that contain many large files and is small on file systems that contain many small files.

The figures under the heading *Free Extents By Size* indicate the totals for free extents of each size. The totals are for free extents of size 1, 2, 4, 8, 16, etc. up to a maximum of the number of data blocks in an allocation unit. The totals are similar to the output of the `df -o` command unless there was recent allocation or deallocation activity (because `fsadm` acts on mounted file systems). These figures provide an indication of fragmentation and extent availability on a file system.

## Extent Reorganization

If the `-e` option is specified, `fsadm` reorganizes data extents on the file system whose mount point is `mount_point`. The primary goal of extent reorganization is to defragment the file system.

To reduce fragmentation, extent reorganization tries to place all small files into one contiguous extent. The `-l` option specifies the size of a file that is considered large. The default is 64 blocks. Extent reorganization also tries to group large files into large extents of at least 64 blocks. Extent reorganizations can improve performance.

Small files can be read or written in one I/O operation, and large files can approach raw-disk performance for sequential I/O operations.

`fsadm` performs extent reorganization on all inodes in the file system. Each pass through the inodes moves the file system closer to optimal organization.

`fsadm` reduces both file fragmentation and free extent fragmentation in each pass. In current versions of VERITAS File System, `fsadm` relies on the VERITAS File System kernel allocation mechanisms to reallocate files in a more favorable

extent geometry. At the same time, the kernel allocation mechanism is prevented from using blocks in areas of the free list that **fsadm** tries to make more contiguous.

The command syntax to perform extent reorganization is:

```
fsadm -e [-E][-s][-v][-l largesize][-p passes][-r rawdev][-t time] mount_point
```

The following example shows the output from the **fsadm -e -s** command:

```
# fsadm -t vxfs -e -s /mnt1
```

### ***Extent Fragmentation Report***

```
Total      Average      Average      Total
Files      File Blks    # Extents    Free Blks
939        11           2            245280
blocks used for indirects: 0
% Free blocks in extents smaller than 64 blks: 8.35
% Free blocks in extents smaller than 8 blks: 4.16
% blks allocated to extents 64 blks or larger: 45.81
Free Extents By Size
          1:  356      2:  292      4:  271      8:  181
          16:  23      32:   7      64:   3     128:   1
          256:   1     512:   0    1024:   1    2048:   1
          4096:  2    8192:  2   16384:  1   32768:  2
Pass 1 Statistics
          Extents  Reallocations  Ioctls          Errors
          Searched  Attempted  Issued  FileBusy  NoSpace  Total
total    2016      1473      789      0         0         0
Pass 2 Statistics
          Extents  Reallocations  Ioctls          Errors
          Searched  Attempted  Issued  FileBusy  NoSpace  Total
total    1836         0         0         0         0         0
```

### ***Extent Fragmentation Report***

```
Total      Average      Average      Total
Files      File Blks    # Extents    Free Blks
939        11           1            245280
blocks used for indirects: 0
% Free blocks in extents smaller than 64 blks: 0.46
% Free blocks in extents smaller than 8 blks: 0.03
% blks allocated to extents 64 blks or larger: 45.53
Free Extents By Size
          1:  10      2:   1      4:   1      8:   4
          16:   3     32:   4     64:   3     128:   3
          256:   3     512:   4    1024:   4    2048:   2
          4096:  3    8192:   1   16384:   1   32768:   2
```

**Note:** In this example, the default five passes were scheduled, but the reorganization completed in two passes.

This file system had a significant amount of free space although there were several free small extents. The situation was corrected by real locating one or more of the extents on many of the files. The files selected for reallocation in this case are those with extents in the heavily fragmented section of the allocation units. The time it takes to complete extent reorganization varies, depending on the degree fragmentation, disk speed, and the number of inodes in the file system. In general, extent reorganization takes approximately one minute for every 100 megabytes of disk space.

In the preceding example, the Extents Searched column contains the total number of extents examined. The Reallocations Attempted column contains the total number of consolidations or merging of extents performed. The ioctls Issued column contains the total number of reorganization request calls made during the pass. This corresponds closely to the number of files that are being operated on in that pass as most files can be reorganized with a single ioctl, (more than one extent may be consolidated in one operation).

The File Busy column (located under the heading Errors) shows the total number of reorganization requests that failed because the file was active during reorganization. The NoSpace column (located under the heading Errors) contains the total number of reorganization requests that failed because an extent presumed free was allocated during the reorganization. The Total column (located under the heading Errors) is the total number of errors encountered during the reorganization and may include errors that were not included with FileBusy or NoSpace.

The following command performs an extent reorganization on all individual files under the mount point mnt1:

```
find /mnt1 -print | fsadm -F vxfs -e -f -
```

### **Files**

lost+found/.fsadm	ock file.
/dev/raw/*	File system devices.
/etc/fstab	Contains static information about file systems.

# Maintenance command: fscdsconv(1M)

## Name

fscdsconv - convert the byte order of a file system

## Availability

VRTSvxfs

## Synopsis

```
fscdsconv [-y|-n] [-r] -f recovery_file special
```

## Description

The fscdsconv utility converts the byte order of the file system specified by special. A file system's metadata is stored in the native byte order of the system on which it was created. To be able to access a file system created on a system with a different byte order, the file system's byte order must first be converted to the native byte order of the system from which it will be accessed.

After starting a conversion and before actually modifying the file system, you will be prompted to confirm the operation. If you choose not to proceed with the operation, the file system will be left unchanged.

The process of converting a file system's byte order can fail for various reasons: system failure, power failure, command failures, user interruption, etc. A temporary file is created by fscdsconv as specified by the -f option so that the file system can be restored to its original condition in the event of a failure. In case of a failure, fscdsconv must be reinvoked with the -r option.

The file system that contains the recovery file must not be a temporary file system that might be cleaned up after a system reboot or whose data integrity is less than the data integrity of the file system being converted. For example, if the file system being converted is on a mirrored volume, then the recovery file should also be on a file system that is capable of tolerating device failures.

The recovery file is not removed when the conversion completes.

See the *VERITAS Storage Foundation Cross-Platform Data Sharing Administrator's Guide* for more information.

## Notes

A backup must be created before performing a conversion.

The file system to be converted must be unmounted before performing the conversion.

If quotas will be used on the file system, before unmounting it, remove the quotas and quotas.grp files. Unmount the file system, then use `fscdsconv` to convert the file system.

On the system to which the file system is being migrated, mount the file system with quotas turned off. Edit the quotas and quotas.grp files to input the usage limits. Turn on quotas for the file system.

### ***Cluster File System Issues***

No cluster issues; command operates the same on cluster file systems.

### **Options**

- f recovery\_file** Specifies the name of the recovery file. Include the full path when specifying the file name.
- n** Assumes a no response to the prompt by `fscdsconv` to continue with the conversion.
- r** Indicates that the `recovery_file` specified by the `-f` option should not be created, but that the existing file should be used to recover the file system after a previous conversion failed.
- y** Assumes a yes response to the prompt by `fscdsconv` to continue with the conversion.

### **Examples**

The following command converts the byte order of the file system `/dev/vx/rdisk/dg1/fs1`, and creates the recovery file `recovery1` on the file system `fs2`:

```
fscdsconv -f /fs2/recovery1 \  
/dev/vx/rdisk/dg1/fs1
```

The following command recovers the file system `/dev/vx/rdisk/dg1/fs1` after a failure, using the recovery file `recovery1` located on the file system `fs2`:

```
fscdsconv -r -f /fs2/recovery1 \  
/dev/vx/rdisk/dg1/fs1
```

### **See also**

`fscdstask(1M)`

*VERITAS Storage Foundation Cross-Platform Data Sharing Administrator's Guide*

# Abbreviations and acronyms

<b>ACL</b>	access control list	<b>VxVM</b>	VERITAS Volume Manager
<b>ASL</b>	Array Support Library		
<b>ASM</b>	array support module		
<b>CDS</b>	cross-platform disk sharing		
<b>DBA</b>	Database administrator		
<b>DMP</b>	Dynamic Multi Pathing		
<b>EFI</b>	Extensible Firmware Interface		
<b>FTP</b>	file transfer protocol		
<b>HCL</b>	Hardware Compatibility List		
<b>HSM</b>	Hierarchical Storage Management		
<b>IBM</b>	International Business Machines Corporation		
<b>ISP</b>	Intelligent Storage Provisioning		
<b>ITSO</b>	International Technical Support Organization		
<b>LSB</b>	least significant byte		
<b>LUN</b>	logical unit number		
<b>MSB</b>	most significant byte		
<b>NFS</b>	network file system		
<b>QoS</b>	Quality of Storage Services		
<b>RAID</b>	redundant array of independent disks		
<b>RCP</b>	Remote copy		
<b>RPM</b>	RPM Package Manager		
<b>SAN</b>	Storage Area Networks		
<b>SCP</b>	Secure Remote copy		
<b>SCSI</b>	Small Computer System Interface		
<b>SSG</b>	Storage Systems Group		
<b>VEA</b>	VERITAS Enterprise Administrator		
<b>VxFS</b>	VERITAS File System		





# 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 about ordering these publications, see “How to get IBM Redbooks” on page 140. Note that some of the documents referenced here may be available in softcopy only.

- ▶ *Tuning SUSE LINUX Enterprise Server on IBM eServer xSeries Servers*, REDP-3862-00
- ▶ *Tuning Red Hat Enterprise Linux on xSeries Servers*, REDP-3861-00
- ▶ *Tuning IBM eServer xSeries Servers for Performance*, SG24-5287-02

## Other publications

These publications are also relevant as further information sources:

- ▶ *VERITAS Storage Foundation 4.0 Installation Guide*
- ▶ *VERITAS Volume Manager 4.0 User's Guide - VERITAS Enterprise Administrator*
- ▶ *VERITAS Volume Manager 4.0 User's Guide*
- ▶ *VERITAS Storage Foundaton 4.0 Cross-Platform Data Sharing Guide*

## Online resources

These Web sites and URLs are also relevant as further information sources:

- ▶ IBM @server xSeries and Linux  
<http://www-1.ibm.com/servers/eserver/linux/xseries/>
- ▶ Tuning for Red Hat Linux  
<http://publib-b.boulder.ibm.com/abstracts/redp3861.html?open>

- ▶ Tuning for SUSE Linux  
<http://publib-b.boulder.ibm.com/abstracts/redp3861.html?Open>
- ▶ Tuning IBM eServer xSeries Servers for Performance  
<http://publib-b.boulder.ibm.com/abstracts/sg245287.html?Open>

## 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](http://ibm.com/redbooks)

## Help from IBM

IBM Support and downloads

[ibm.com/support](http://ibm.com/support)

IBM Global Services

[ibm.com/services](http://ibm.com/services)

# Index

## Numerics

360 Commerce 3

## A

access control list (ACL) 13, 29  
access time 11  
ACCPAC 3  
ACL. See access control list (ACL).  
active files 21  
AIX 26, 28  
Apache 2  
application 28–29  
application migration 10  
application usage 10  
array subsystem 15  
array support library 42, 84  
Array Support Library (ASL) 32  
Array Support Module (ASM) 22  
ASCII text file 10  
Axiom 3

## B

BaaN Invensys 3  
backup 15  
bandwidth 21  
BEA 6  
big endian 14  
bin group 14  
binary file 10  
block boundary issues 27  
block level 20  
block size 27  
blocks 130  
BMC 3  
boot disk 26  
Brio 3  
Bynari 3  
byte order 10

## C

C programming language 12  
cache space 21

Cadence 3  
Caldera 3  
cdsdisk 27  
cdsdisk disk type 27  
CDS-enabled disks 26  
Checkpoint 3  
CISCO 3  
cluster mounted file system 124  
command line arguments 22  
command script 13  
compute intensive environment 21  
Compuware 3  
concatenated disks, 62  
CPU 71  
CPU architecture 10  
CPU usage 69  
cross-platform data sharing disk format 27  
Crystal Decisions 3

## D

data 21, 28  
data collection 69  
data item 12  
data length 15  
data migration 11, 13, 80–81  
data migration performance 68  
data protection 5  
data structure 11–12, 15  
data transfer 92  
data type 15  
data type definition 12  
database design 15  
database migration 11  
DB2 8.1 118  
DB2 high-end template 22  
DB2 UDB 2  
db2look 118  
db2move 118  
design 10  
development tools 2  
device entry 28  
device quota 28  
Dimensional Insight 3

- directory 11
- directory depth length 80
- directory fragmentation 128
- directory hierarchy 12
- directory reorganization 129
- disk blocks 13
- disk label 26
- disk management 10
- disk partition table 27
- disk space 12, 15
- disk transfer speed 67
- disk utilization 18
- diskgroup 27, 29, 83
- disks 27
- DMP 32
- Domino 3
- DS4300 storage subsystem 31
- Dynamic Multi Pathing 18, 84
- Dynamic Multi Pathing (DMP) 32

## E

- EFI disks 26
- endian 29, 82
- endian dependent format 28
- endianess 10, 14
- enterprise journaling file system 19
- enterprise storage solution 18
- eOneGroup 3
- Everyplace 2
- exp utility 101

## F

- failover driver 33
- FAStT Storage Manager 83
- FAStT Storage Manager 8.4 31
- FAStT600 31
- Fibre Channel 32
- Fibre Channel host bus adapter 32
- file 11
- file management 13
- file migration 10
- file name 12
- file name length 80
- file owner 21
- file ownership 14
- file placement 12
- file pointer table 11
- file size 10–11, 21, 80

- file system 12, 28
- file system size 11, 80
- file type 11
- Firewall 3
- FlashSnap 23, 92
- file structure 2
- fragmentation 127, 131
- fsadm 123
- fscdsconv 82
- fscdstask 80
- fscdstask validate 80

## G

- group ID 80

## H

- hardware architecture 26
- hardware management 10
- HBA device driver 84
- head command 33
- header files 12
- HP 28
- HP-UX 26
- HS20 31
- human language 22
- Hyperion 3

## I

- I/O bandwidth 71
- I/O collection 75
- I/O patterns 22
- I/O statistics 69
- I/O usage 69
- Informix 2
- inode 10–11
- inode number 11
- installation shell script 33
- Instant Snapshots 18
- Intelligent Markets 3
- Intelligent Storage Provisioning (ISP) 51
- iostat 69
- IP network 5

## J

- JD Edwards CRM for Financial Services 3

## **K**

kernel 2, 10

## **L**

least significant byte (LSB) 14  
Legato 3  
libc 12  
life cycle 22  
linear curve 70  
Linux 2, 5–6, 10–11, 13, 19, 25, 27, 31  
Linux host group 96  
Linux kernel 11  
Linux Redhat 118  
little endian 14  
log 21  
logical drive mapping 96  
logical names 18  
Logical Unit Number  
    See LUN  
logical volume managers 26  
logsize 126  
low latency 81  
LSB. See least significant byte (LSB).  
LSDC 3  
LUN 20, 111

## **M**

Ma 123  
MarCole 3  
memory management 10  
Mentor Graphics 3  
metadata 15, 21, 28, 68, 70, 81  
metadata conversion process 28  
Microsoft Windows 29  
migration 10  
migration challenges 10  
migration objects 10  
most significant byte (MSB) 14  
MP3 files 29  
mpstat 69  
MSB. See most significant byte (MSB).  
multiple host mapping 84  
multiple host types 84  
mySAP.com 3

## **N**

network bandwidth 15

network connection 86  
network file system (NFS) 15  
network topology 15  
NFS. See network file system (NFS).

## **O**

offline conversion 28  
on-disk format 28  
operating system 10  
Oracle 2, 6, 118  
Oracle application migration 100

## **P**

partition table 27  
part-linear scalability 70  
PeopleSoft 6  
performance 15  
permission 11  
physical device addresses 18  
physical disk 18  
platform block 26  
point-in-time copy technology 23  
PolyServe 3  
Portable Data Containers ix, 16, 19, 24–26, 29, 32,  
62, 67, 69, 73, 76, 79–82, 88, 91, 100–101, 105,  
118, 150  
POWER 14  
private region 27  
programming language 10  
public region 27

## **Q**

QAD 3  
QLLogic Fibre Channel 31  
Quality of Storage Service 21  
Quality of Storage Services 19–20  
quota 13

## **R**

RAID 18, 20, 22  
Rational 2  
RCP. See remote copy (RCP).  
reconnection 16  
recovery file 81  
Redbooks Web site 140  
    Contact us xi  
Relavis 3

remote copy (RCP) 15  
replication 15  
RISC architecture 14  
rpms 36, 39  
Rueters 3

## S

Sage 3  
Samba 3  
SAN connection 83  
Sanchez 3  
SAP 6  
SAP BI 3  
scalability 69  
SCP. See secure remote copy (SCP).  
SCSI mode-sense commands 26  
secure remote copy (SCP) 15  
Sendmail 3  
shared storage subsystem 85  
shell 2  
shell script 33  
Siebel 6  
small endian 28  
snapshot volume 84, 93, 96  
Solaris 11, 25, 31, 80, 118  
space-optimized snapshots 23  
SPARC 14  
split diskgroup 94  
SQL 119  
SQL script 101  
Steeleye 3  
storage administrator 28  
storage architect 22  
storage area networks (SAN) 21  
storage array 18  
Storage Foundation rpms 37  
storage hierarchy 20  
storage management 17  
storage partitioning feature 83  
storage templates 23  
Sun 28, 80  
Sun Ultra-60 31  
SUNGARD 3  
SUSE eMail Server 3  
SUSE Firewall 3  
svrmgrl prompt 101  
Sybase 2  
Symantec 3

Synopsis 3  
system call 13  
system libraries 12

## T

tablespace 102  
tape library 15  
task management 10  
telnet 33  
throughput results 74  
throughput values 77  
time 69  
Tivoli 3  
tree structure 12  
Trustix 3

## U

UltraSparc 71  
UNIX 2, 10, 27  
UNIX file system 11  
UNIX permissions 13  
updated time 11  
User ID 80  
user ID limitation 80  
utilities 2

## V

VEA. See VERITAS Enterprise Administrator.  
VERITAS  
    Array Support Library 32  
    Cluster Server 4, 6  
    Enterprise Administrator (VEA) 18, 31, 44, 48,  
    84, 92  
    File System 4, 11–12, 18, 21, 28  
    File System on Solaris 80  
    FlashSnap 23, 83  
    Hardware Compatibility List 43  
    i3 APM software 6  
    i3 for Linux 6  
    Indepth 6  
    Inform 6  
    Insight 6  
    Intelligent Storage Provisioning 22  
    Multi Volume File System 21  
    NetBackup 4  
    NetBackup for Linux 5  
    Opforce for Linux 6

Portable Data Containers. See Portable Data Containers.  
Storage Foundation 4, 17, 83  
Storage Foundation Cluster File System 5  
Storage Foundation for Database 4  
Storage Foundation for Linux 4  
Storage Foundation for Oracle RAC 4–5  
technote 260741 42  
Volume Manager 18, 26, 42, 97  
Volume Manager Dynamic Multi Pathing 22  
Volume Replicator 4  
Volume Replicator for Linux 5  
virtualization 18  
VMware 3  
volume set 20  
VRTSvxfs 123  
vxdg list 64  
vxfs file system type 57  
vxstat 69  
VxVM ID block 27

## **W**

WebSphere Application Server 2  
WebSphere Commerce Suite 3  
white space 18













**Redbooks**

# Migrating to Linux Using VERITAS Portable Data Containers on IBM eServer xSeries and BladeCenter

**Fast migration of  
enterprise data from  
UNIX to Linux**

**Mechanics of  
Portable Data  
Containers and  
step-by-step  
migration from  
Solaris to Linux**

**Best practices for  
migrating your data  
to Linux**

This IBM Redbook discusses Portable Data Containers. Portable Data Containers are a tool that allows customers to migrate their applications from UNIX to Linux more seamlessly. They are a part of VERITAS Storage Foundations 4.0. With Portable Data Containers, a customer can migrate an application without moving that application's data.

This redbook describes the concepts of storage management and the key features of VERITAS Storage Foundations 4.0 and . In addition, it discusses a number of considerations that you should review prior to a UNIX to Linux migration, and then takes you through a step-by-step migration from Solaris to Linux. It discusses the performance and scalability of Portable Data Containers and compares the throughput of Portable Data Containers with theoretical network throughput and disk transfer speeds. Finally, this redbook includes best practices for migrating applications to Linux using Portable Data Containers. In particular, it identifies the steps that are required to migrate an Oracle or DB2 database from a UNIX server to a Linux server. With this knowledge, customers can accelerate their migration plans to Linux to take advantage of the cost benefits provided by the open-source operating system.

The audience for this redbook are IBM Clients and Business Partners as well as IT specialists and database administrators.

**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](http://ibm.com/redbooks)**