Data Migration to IBM Disk Storage Systems

Highlights tool and techniques for Open Systems and z/OS

Addresses appliance, host, and storage-based migrations

Includes z/OS TDMF, TDMF TCP/IP, and zDMF

Chris Seiwert
Peter Klee
Lisa Martinez
Max Pei
Mladen Portak
Alex Safonov
Edgar Strubel
Gabor Szabo
Ron Verbeek

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

Second Edition (February 2012)

This edition applies to data migration products and techniques as of July 2011.

© Copyright International Business Machines Corporation 2011, 2012. 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

Notices ......................................................... xi  
Trademarks ................................................. xii  

Summary of changes ............................................ xiii  
February 2012, Second Edition ............................... xiii  

Preface ......................................................... xv  
The team who wrote this book ................................ xv  
Now you can become a published author, too! .............. xvii  
Comments welcome ............................................ xviii  
Stay connected to IBM Redbooks .............................. xviii  

Part 1. Data migration ............................................ 1  

Chapter 1. Introducing disk data migration ...................... 3  
1.1 Data migration and its business reasons .................. 4  
1.2 Aspects of data migration ................................ 4  
1.3 Addressing data migration issues ......................... 5  

Chapter 2. Migration techniques and processes ................... 7  
2.1 Data migration techniques ................................ 8  
2.1.1 Host-Based migration ................................. 8  
2.1.2 Array-Based migrations ............................... 10  
2.1.3 Appliance-Based migrations ............................ 10  
2.1.4 Migrating using backup and restore .................. 10  
2.1.5 Data migration in System z environments ............. 11  
2.1.6 Summary .............................................. 13  
2.2 Key decision factors ...................................... 14  
2.2.1 Migration performance ............................... 14  
2.2.2 Migration backout scenarios, protecting source and target volumes .................. 14  
2.2.3 Understand target storage requirements ............. 15  
2.2.4 Different source and target hardware ............... 16  
2.2.5 Application downtime ................................ 17  
2.2.6 Key decision factors summary ....................... 18  
2.3 Data migration process .................................... 18  
2.3.1 Planning phase ....................................... 19  
2.3.2 Validation phase ..................................... 27  
2.4 Preparing DS8000 for data migration ..................... 27  
2.4.1 Available hardware resources ....................... 28  
2.4.2 Understanding array characteristics ................. 28  

Chapter 3. IBM service offerings .................................. 31  
3.1 IBM Global Services offerings ............................ 32  
3.1.1 IBM Migration Services for data for Open Systems ........ 32  
3.1.2 IBM Migration Services for System z data ............ 33  
3.1.3 IBM Migration Services for Network Attached Storage Systems .................. 35  
3.2 IBM Systems and Technology Group Lab Services ........ 36  
3.2.1 IBM XIV Migration Services ....................... 36  
3.2.2 IBM DS8000 Data Migration Services ............... 37
3.2.3 Additional Services & Offerings .................................................. 38

Part 2. Migration scenarios ............................................................. 39

Chapter 4. Data migration using IBM Remote Mirror and Copy .................. 41
4.1 IBM System Storage Remote Mirror and Copy overview ..................... 42
4.1.1 Metro Mirror ........................................................................ 43
4.1.2 Global Copy ....................................................................... 44
4.1.3 Target DS8000 configuration considerations ............................ 46
4.1.4 Automation software: Tivoli Storage Productivity Center for Replication 47
4.2 DS8000 user interface for Copy Services ...................................... 47
4.2.1 IBM System Storage DS8000 Storage Manager ...................... 47
4.2.2 IBM System Storage DS8000 command-line interface .............. 49
4.2.3 System z interfaces ............................................................ 49
4.3 Data migration from Enterprise Storage Server Model 800 to DS8000 .... 51
4.3.1 Adding Enterprise Storage Server Copy Services Domain to the DS8000 Storage Complex ......................................................... 51
4.3.2 Creating Remote Mirror and Copy paths ................................ 54
4.3.3 Creating Remote Mirror and Copy pairs ................................ 64
4.3.4 Completing the data migration ............................................. 72
4.4 Data migration from DS8000 to DS8000 ....................................... 81
4.4.1 Configuring the remote DS8000 .......................................... 82
4.4.2 Creating Remote Mirror and Copy paths .............................. 83
4.4.3 Creating Remote Mirror and Copy pairs .............................. 85
4.4.4 Completing the data migration from DS8000 to DS8000 .......... 85
4.5 Post-migration tasks ................................................................. 90
4.5.1 AIX post-migration tasks ................................................... 90
4.5.2 Solaris and VxVM post-migration tasks ................................ 92
4.5.3 System z post-migration tasks ............................................. 99

Chapter 5. DSCLIbroker ....................................................................... 105
5.1 Overview ................................................................................. 106
5.1.1 DSCLIbroker concepts ....................................................... 106
5.1.2 DSCLIbroker scripts .......................................................... 113
5.1.3 User customized scripts ..................................................... 115
5.1.4 Additional useful scripts .................................................... 118
5.2 System environment ............................................................... 118
5.2.1 Architecture ..................................................................... 118
5.2.2 Communication .................................................................. 119
5.2.3 Users ................................................................................. 120
5.2.4 DSCLIbroker maintenance ................................................. 120
5.2.5 Licenses ............................................................................. 121
5.3 Programming techniques ......................................................... 121
5.3.1 Using DSCLIbroker scripts ................................................. 121
5.3.2 Using DSCLIbroker libraries .............................................. 122
5.4 Automation techniques ........................................................... 125
5.4.1 Subjects for writing automation scripts ................................ 125
5.4.2 Generating DSCLI scripts .................................................. 126
5.4.3 Using control files ............................................................ 127
5.5 Considerations about configuration data ...................................... 127
5.5.1 Data sources ..................................................................... 127
5.5.2 Importing data to the repository ........................................ 128
5.6 Example migration using DSCLIbroker ....................................... 128
5.6.1 Overview ......................................................................... 128
## Chapter 6. IBM XIV data migration

6.1 Overview ................................................................. 134
6.2 Handling I/O requests .................................................. 135
  6.2.1 Methods of handling write requests ................................. 135
  6.2.2 Multi-pathing with data migrations ................................. 137
6.3 Data migration steps ................................................... 138
  6.3.1 Initial connection setup ............................................. 139
  6.3.2 Creating a data migration volume on XIV .......................... 144
  6.3.3 Activate a data migration on XIV .................................. 148
  6.3.4 Defining the host on XIV and bringing the host online .......... 149
  6.3.5 Completing the data migration on XIV ............................ 150
6.4 Command-line interface ............................................... 152
  6.4.1 Using XCLI scripts or batch files ................................. 155
  6.4.2 Sample scripts ..................................................... 156
6.5 Manually creating the migration volume ........................... 157
6.6 Changing and monitoring the progress of a migration ............ 158
  6.6.1 Changing the synchronization rate ................................ 159
  6.6.2 Monitoring migration speed ....................................... 160
  6.6.3 Monitoring migration using the XIV event log .................... 160
  6.6.4 Monitoring migration speed through the fabric ................ 161
  6.6.5 Monitoring migration speed through the non-XIV storage ...... 161
6.7 Thick-to-thin migration .............................................. 162
  6.7.1 Writing zeros to recover space ................................... 162
  6.7.2 Recovering space after the migration ............................. 163
6.8 Resizing the XIV volume after migration .......................... 163
6.9 Troubleshooting ....................................................... 165
  6.9.1 Target connectivity fails ......................................... 165
  6.9.2 Remote volume LUN is unavailable ............................... 166
  6.9.3 Local volume is not formatted .................................... 167
  6.9.4 Host server cannot access the XIV migration volume .......... 168
  6.9.5 Remote volume cannot be read ................................... 168
  6.9.6 LUN is out of range .............................................. 168
6.10 Backing out of a data migration ................................... 169
  6.10.1 Back out before migration is defined on the XIV ............. 169
  6.10.2 Back out after a data migration is defined but not activated .. 169
  6.10.3 Back out after a data migration is activated but is not complete ... 169
  6.10.4 Back out after a data migration reaches the synchronized state ... 170
6.11 Migration checklist ................................................ 170
6.12 Other considerations ............................................... 173

## Chapter 7. SAN Volume Controller-based migration

7.1 IBM System Storage SAN Volume Controller overview ............ 176
  7.1.1 SAN Volume Controller components, concepts, and terminology .. 177
  7.1.2 SAN Volume Controller copy services ............................ 183
  7.1.3 Metro Mirror ..................................................... 183
  7.1.4 Global Mirror .................................................... 185
7.2 SAN Volume Controller concepts for migrating the data ........ 185
Chapter 8. Using mirroring techniques

7.2.1 Data migration using SAN Volume Controller volume migration .................................. 185
7.2.2 Data migration using SAN Volume Controller FlashCopy ............................................. 187
7.2.3 Data migration using SAN Volume Controller Metro Mirror ........................................ 188
7.2.4 Data migration using mirrored volumes ........................................................................ 189
7.3 Migrating using SAN Volume Controller ........................................................................... 190
7.3.1 Migrating extents ............................................................................................................ 191
7.3.2 Migrating extents off an MDisk that is being deleted ....................................................... 192
7.3.3 Migrating a volume between storage pools ................................................................... 192
7.3.4 Using volume mirroring ................................................................................................. 193
7.3.5 Image mode volume migration ....................................................................................... 194
7.3.6 Migrating the volume to image mode ............................................................................. 196
7.3.7 Migrating a volume between I/O groups ........................................................................ 197
7.3.8 Monitoring the migration progress .................................................................................. 198
7.3.9 Parallelism ...................................................................................................................... 199
7.3.10 Error handling ............................................................................................................. 199
7.3.11 Migration tips ............................................................................................................... 200
7.4 SAN Volume Controller Migration preparation prerequisites .............................................. 200
7.4.1 Fabric zoning ................................................................................................................ 200
7.4.2 Connect SAN Volume Controller to the fabric for migration ........................................ 201
7.4.3 Remove SAN Volume Controller from the fabric after migration ................................. 202
7.4.4 Back-End storage consideration ..................................................................................... 202
7.4.5 Unsupported storage systems ........................................................................................ 203
7.4.6 Host attachment ............................................................................................................ 203
7.5 Migrating SAN disks to SAN Volume Controller volumes and back to SAN ................. 203
7.5.1 Connecting the SAN Volume Controller to your SAN fabric ........................................ 205
7.5.2 Preparing your SAN Volume Controller to virtualize disks ......................................... 206
7.5.3 Moving the LUNs to the SAN Volume Controller ......................................................... 210
7.5.4 Migrating image mode volumes to volumes .................................................................. 213
7.5.5 Performance analysis ..................................................................................................... 219
7.5.6 Preparing to migrate from the SAN Volume Controller ................................................ 220
7.5.7 Creating new LUNs ........................................................................................................ 222
7.5.8 Migrating the managed volumes .................................................................................... 223
7.5.9 Removing the LUNs from the SAN Volume Controller .................................................. 225
7.6 SAN Volume Controller Volume migration between two storage pools ........................... 227
7.6.1 Environment description ............................................................................................... 228
7.6.2 Performance measurement ............................................................................................. 232
7.6.3 Migration steps ............................................................................................................. 232
7.6.4 Performance Analyses ................................................................................................... 235
7.7 Data migration using SAN Volume Controller mirrored volumes .................................... 237
7.7.1 Environment description ............................................................................................... 237
7.7.2 Creating mirrored volumes using the SAN Volume Controller GUI ............................... 241
7.7.3 Creating mirrored volumes using CLI .......................................................................... 244
7.7.4 Performance analysis ..................................................................................................... 246
7.8 Data migration using SAN Volume Controller Metro Mirror ........................................... 246
7.8.1 SAN Volume Controller Metro Mirror partnership ....................................................... 247
7.8.2 SAN Volume Controller Metro Mirror relationships ..................................................... 249
7.8.3 Starting and monitoring SAN Volume Controller Metro Mirror Copy .......................... 254
7.8.4 Stopping SAN Volume Controller Metro Mirror Copy .................................................. 257
7.8.5 Performance overview ................................................................................................... 261
7.9 SAN Volume Controller as data migration engine ............................................................. 262
7.10 Other resources ............................................................................................................... 263

Chapter 8. Using mirroring techniques..................................................................................... 265
11.2.1 Keeping current ................................................. 488
11.2.2 Creating a consistent data structure ......................... 489
11.2.3 Summary of replication tasks and steps ....................... 490
11.2.4 Link performance information ............................... 494
11.3 TDMF TCP/IP example replication .............................. 496
11.3.1 Overview of replication ..................................... 497
11.3.2 Naming conventions ......................................... 497
11.3.3 TDMF TCP/IP examples for a network test .................. 507
11.3.4 TDMF TCP/IP performance example ......................... 512
11.3.5 TCP/IP information ......................................... 512

Appendix A. Network block devices for Linux ......................... 515
Network block devices ................................................. 516
Suggested methods ..................................................... 516
Using software RAID1 ................................................. 517
Summary ................................................................. 518

Appendix B. CLI Conversion ........................................... 521
Migrating ESS Copy Services tasks to DS8000 CLI .................. 522
Review the ESS tasks to migrate ..................................... 522
Convert the individual tasks ......................................... 526
Copy Services commands .............................................. 529
Copy Services notes .................................................... 530

Related publications .................................................. 531
IBM Redbooks .......................................................... 531
Other publications ...................................................... 531
Online resources ........................................................ 531
Help from IBM .......................................................... 531

Index ................................................................. 533
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 websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

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

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

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

COPYRIGHT LICENSE:

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

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

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

<table>
<thead>
<tr>
<th>IBM®</th>
<th>AIX®</th>
<th>BladeCenter®</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM®</td>
<td>CICS®</td>
<td>DB2®</td>
</tr>
<tr>
<td>IBM®</td>
<td>DS4000®</td>
<td>DS6000™</td>
</tr>
<tr>
<td>IBM®</td>
<td>Notes®</td>
<td>Easy Tier®</td>
</tr>
<tr>
<td>IBM®</td>
<td>OS/390®</td>
<td>Enterprise Storage Server®</td>
</tr>
<tr>
<td>IBM®</td>
<td>PowerHA®</td>
<td>ESCON®</td>
</tr>
<tr>
<td>IBM®</td>
<td>PR/SM™</td>
<td>eServer™</td>
</tr>
<tr>
<td>IBM®</td>
<td>pSeries®</td>
<td>FICON®</td>
</tr>
<tr>
<td>IBM®</td>
<td>RACF®</td>
<td>FlashCopy®</td>
</tr>
<tr>
<td>IBM®</td>
<td>Rational®</td>
<td>GDPS®</td>
</tr>
<tr>
<td>IBM®</td>
<td>Redbooks®</td>
<td>Global Technology Services®</td>
</tr>
<tr>
<td>IBM®</td>
<td>Redbooks (logo)®</td>
<td>HACMP™</td>
</tr>
<tr>
<td>RMF™</td>
<td>S/390®</td>
<td>Tivoli®</td>
</tr>
<tr>
<td>S/390®</td>
<td>System Storage DS®</td>
<td>VM/ESA®</td>
</tr>
<tr>
<td>System i®</td>
<td>System Storage®</td>
<td>z/OS®</td>
</tr>
<tr>
<td>System z®</td>
<td>TDMF®</td>
<td>z/VSE®</td>
</tr>
<tr>
<td>zSeries®</td>
<td>zSeries®</td>
<td></td>
</tr>
</tbody>
</table>

The following terms are trademarks of other companies:

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

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

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

Disk Magic, and the IntelliMagic logo are trademarks of IntelliMagic BV in the United States, other countries, or both.

QLogic, and the QLogic logo are registered trademarks of QLogic Corporation. SANblade is a registered trademark in the United States.

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

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

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

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

Summary of Changes
for SG24-7432-01
for Data Migration to IBM Disk Storage Systems
as created or updated on February 23, 2012.

February 2012, Second Edition

This revision reflects the addition, deletion, or modification of new and changed information described below.

New information
► Data migration in IBM PowerHA® clustered environments
► IBM XIV® data migration
► IBM® DSCLIbroker scripting framework
► IBM TDMF® TCP/IP for IBM z/OS®

Changed information
► Screen captures updated to reflect latest software available at the time of writing
Preface

Data migration has become a mandatory and regular activity for most data centers. Companies need to migrate data not only when technology needs to be replaced, but also for consolidation, load balancing, and disaster recovery.

This IBM Redbooks® publication addresses the aspects of data migration efforts while focusing on the IBM System Storage® as the target system. Data migration is a critical and complex operation, and this book provides the phases and steps to ensure a smooth migration. Topics range from planning and preparation to execution and validation.

The book also reviews products and describes available IBM data migration services offerings. It explains, from a generic standpoint, the appliance-based, storage-based, and host-based techniques that can be used to accomplish the migration. Each method is explained including the use of the various products and techniques with different migration scenarios and various operating system platforms.

This document targets storage administrators, storage network administrators, system designers, architects, and IT professionals who design, administer or plan data migrations in large data Centers. The aim is to ensure that you are aware of the current thinking, methods, tools, and products that IBM can make available to you. These items are provided to ensure a data migration process that is as efficient and problem-free as possible.

The material presented in this book was developed with versions of the referenced products as of June 2011.

The team who wrote this book

This IBM Redbooks publication was produced by a team of specialists from around the world working at the International Technical Support Organization, San Jose Center.

Chris Seiwert is an IT Specialist in the IBM European Storage Competence Center (ESCC), Mainz, Germany. He has 12 years of experience in SAN, High End Storage, Data Center Analysis and Planning, and 19 years in computer science. He holds a Bachelor of Science degree in Computer Science, and his areas of expertise also include Data Migration, JavaTM development, and HA Cluster solutions. Chris filed his first patent in 2005 for an IBM Data Migration tool. He has written about SAN and data migration in IBM internal and external publications, and has delivered presentations and workshops about these topics. He has co-authored three previous IBM Redbooks and was the ITSO lead for this book.

Peter Klee is an IBM Professional Certified IT specialist in IBM Germany. He has 17 years of experience in Open Systems platforms, SAN networks, and high end storage systems in huge data centers. He formerly worked in a large bank in Germany where he was responsible for the architecture and the implementation of the disk storage environment. This environment has included the installation from various storage vendors, including EMC, HDS, and IBM. He joined IBM in 2003, where he worked for Strategic Outsourcing. Since July 2004, he has worked for ATS System Storage Europe in Mainz. His main focus is Copy Services, Disaster Recovery, and storage architectures for IBM DS8000® in the open systems environment.

Lisa Martinez is a storage architect in the Speciality Services Area in GTS. She has been in this position since the beginning of 2011. In 2010 Lisa took a temporary assignment as a Global Support Manager for Cardinal Health. Prior to this assignment she was a test architect.
in disk storage focusing on system level test for XIV for three years, and copy services for DS8K. Lisa holds degrees in Computer Science from New Mexico Highlands University and Electrical Engineering from the University of New Mexico. She has been employed with IBM for 14 years.

Max Pei is an Infrastructure Architect for GTS in IBM Canada, specializing in SAN, Storage, and Backup systems. He has 14 years of experience in the IT industry and has been with IBM since 2008. He holds a degree in Metallurgical Engineering. His areas of expertise include planning and implementation of midrange and enterprise storage, storage networks, backup systems, data migration and server virtualization.

Mladen Portak is a Client Technical Storage Specialist for STG in IBM Croatia, and specializes in Storage systems. He is certified on the IBM Midrange and Enterprise Storage Systems and Microsoft MCP. Mladen has 16 years of experience in the IT industry and has been with IBM since 2008. Before joining IBM he worked on customer side as a team leader responsible for virtualization solutions. His current area of expertise includes the planning, architecture and implementation of midrange and enterprise storage, and storage area networks for Open Systems.

Alex Safonov is a Senior IT Specialist with System Sales Implementation Services, IBM Global Technology Services® Canada. He has over 20 years of experience in the computing industry, with the last 15 years spent working on Storage and UNIX solutions. He holds multiple product and industry certifications, including IBM Tivoli® Storage Manager, IBM AIX®, and SNIA. Alexander spends most of his client contracting time working with Tivoli Storage Manager, data archiving, storage virtualization, and replication and migration of data. He holds an M.S. Honors degree in Mechanical Engineering from the National Aviation University of Ukraine.

Edgar Strubel is a Server and Storage Specialist with the STG LAB Services Europe, Mainz, Germany. He started working at IBM at the end of 2000 (Mainz Briefing Center, ATS Team) and since 2002 he has been involved in online data migration in IBM zSeries® environments. Prior to joining IBM, starting in 1980, Edgar worked at BASF/COMPAREX in mainframe IT environments, performing service and support in hardware and software for printers, tapes, libraries, disks, and processors.

Gabor Szabo is an IBM Certified Solution Designer working as Test Engineer team leader of DS8000 Development Support team in IBM Hungary, Vac. He has 8 years of experience in the IT industry and has been with IBM since 2005. Gabor current area of expertise includes the high-end storage system testing, test optimization, and new product implementation.

Ron Verbeek is a Senior Consulting IT Specialist with Data Center Services, IBM Global Technology Services Canada. He has over 23 years of experience in the computing industry with IBM, with the last 11 years working on Storage and Data solution services. He holds multiple product and industry certifications, including SNIA Storage Architect, and has co-authored one previous IBM Redbooks on the IBM XIV Storage Subsystem. Ron spends most of his client time in technical pre-sales solutions, defining and creating storage and optimization solutions. He has extensive experience in data transformation services and information life cycle consulting. He holds a Bachelor of Science degree in Mathematics from McMaster University in Canada.
Figure 1   The team who wrote this book

From left to right: Bert Dufrasne, Edgar Strubel, Chris Seiwert, Alex Savonov, Ron Verbeek, Peter Klee, Mladen Portak, Gabor Szabo, Max Pei. Missing: Lisa Martinez.

Special thanks to Uwe Heinrich Mueller, Uwe Schweikhard, and Mike Schneider IBM Lab services Mainz, for excellent lab support during the residency.

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

Bernd Mueller, Michael Murtagh, Ralf Wohlfarth
IBM Systems & Technology Group Mainz, Europe

Lori Bideaux and Bert Dufrasne
IBM International Technical Support Organization (ITSO)

Michael Moss
Technical Support Softek (an IBM Company)

Now you can become a published author, too!

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

Find out more about the residency program, browse the residency index, and apply online at: ibm.com/redbooks/residencies.html
Comments welcome

Your comments are important to us!

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

► Use the online Contact us review Redbooks form found at:
  ibm.com/redbooks

► Send your comments in an email to:
  redbooks@us.ibm.com

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

Stay connected to IBM Redbooks

► Find us on Facebook:
  http://www.facebook.com/IBMRedbooks

► Follow us on Twitter:
  http://twitter.com/ibmredbooks

► Look for us on LinkedIn:
  http://www.linkedin.com/groups?home=&gid=2130806

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

► Stay current on recent Redbooks publications with RSS Feeds:
  http://www.redbooks.ibm.com/rss.html
Part 1

Data migration

This part defines data migration and reviews the various business reasons for migrating data. It explores the advantages and disadvantages of the various migration techniques, and highlights some of the available migration products. It also includes the phases of a migration project and available IBM disk migration services.

This part contains the following chapters:
- Introducing disk data migration
- Migration techniques and processes
- IBM service offerings
Introducing disk data migration

This chapter defines disk data migration and examines the business reasons for migrating data. It contains the following sections:

- Data migration and its business reasons
- Aspects of data migration
- Addressing data migration issues
1.1 Data migration and its business reasons

Data migration is the transferring of data between storage subsystem types, formats, or computer systems. These systems can be single or multiple, and similar or dissimilar. Migration is usually performed under program control to try to make movement of data as complete and as automated as is possible.

Data migration is needed when you change computer systems or upgrade to new products. As with any data center change, you want to avoid disrupting or disabling active applications.

You might migrate data for the following reasons:
- Your organization wants to use new storage devices to increase the size of your online storage or for more flexibility.
- You want to resolve performance issues.
- You need to physically reduce the footprint of your storage subsystem within the data center and release space. Reducing the footprint reduces the costs of power consumption, air conditioning, and lighting.
- You want to employ data migration technologies to implement a disaster recovery solution.
- You need the new functions and facilities offered by evolving technology to stay competitive.
- You need to relocate your data center.

1.2 Aspects of data migration

Migrating data has the following general concerns, which require that you take a holistic view of your data center:
- Configuration: New hardware might need to be configured into the systems being migrated before you can start.
- Operations: A time needs to be found when the migration can be performed. Scheduling is especially important if you use offline data migration techniques that require the applications to be stopped. After the migration, your data center operating procedures will need to be reviewed and updated.
- Data: Not all data can be migrated in the same way. For example, local paging data sets on IBM OS/390® cannot be moved directly. The restriction is because manual handling of local paging data sets is required to keep OS/390 systems up and running.
- Infrastructure: Changes to infrastructure might be required to accommodate new storage subsystem technologies. For example, new technology can require new attachment methods or enhanced data rate capabilities. The most common upgrades are from Escon to Ficon attachment, and for Open Systems installing a SAN Fabric or upgrading your existing fabric.
- Applications: You might need to modify or add installation parameters in your applications to accommodate new storage subsystem devices or infrastructures.
- Performance: Migration of data is an opportunity to maximize performance by carefully relocating high use data. However, if you are not careful about where you migrate data, you can create performance issues.
1.3 Addressing data migration issues

All of the above scenarios are fairly commonplace in modern IT operations and in virtually all businesses. However, even these routine practices can cause problems for IT managers, database administrators, and their teams. The problems tend to include the following issues:

- Extended or unexpected downtime
- Data corruption, missing data, and data loss
- Application performance issues
- Technical compatibility issues

To avoid impact to business operations, most data migration projects are scheduled during off-hours, primarily during weekends and over extended holiday periods. However, this delay can increase migration costs because of staff overtime and negatively impact staff morale. Furthermore, taking systems down for migration, even over a weekend period, can severely affect business operations. The impact is especially severe if you cannot bring the systems back up in time for the online day.

These potential problems cause some organizations to significantly delay purchasing new technology, or even to delay the deployment of already purchased technology. These delays cause further problems because older hardware can require more hands-on maintenance, generally has lower performance, and is inherently more prone to failure.

You buy and deploy new technology to eliminate these issues. Therefore delays in implementing new technology increases business risk because of your need to run around-the-clock 24x7 applications with ever-shrinking batch windows. In addition, delaying deployment of an already purchased or leased storage device raises its effective cost because you are paying for both old and new devices.

Data migration can be low risk. Current IBM data migration technologies allow you to perform most migrations with no downtime. IPL or server restarts are not always required, and no volumes need to be taken offline. However, you might have to perform a scheduled IPL or restart of a server if you are adding new equipment or applying system maintenance. In addition, the latest migration software tools allow nondisruptive migration, allowing applications to remain online during data movement without significant performance delays.
Migration techniques and processes

This chapter reviews the data migration techniques and highlights how to select the appropriate technique.

The chapter also includes a summary of the migration process based on a three-phase approach:

- Planning phase
- Migration phase
- Validation phase

This chapter includes the following sections:

- Data migration techniques
- Key decision factors
- Data migration process
- Preparing DS8000 for data migration
2.1 Data migration techniques

Migrating data is always a disruptive process. Every migration technique affects the normal operations of the system.

Selecting the appropriate technique depends on the criticality of the data being moved, the resources available, and other business constraints and requirements. The different techniques have different risks. Select the technique that provides the best combination of efficiency of migration and low impact to system and users.

Various software products and tools can be used to migrate data:

- Volume management products
- Host-Based replication products
- Array-Based replication products
- Relocation utilities
- Custom-Developed scripts

Each product or tool has strengths and weaknesses. These include impact on performance, operating system support, storage vendor platform support, and whether application downtime is required to migrate data. Select the best product or tool for your needs.

Determine whether you can migrate data online or offline:

- Online migrations: Data can be copied from one set of volumes to another with no impact to users and applications. In most cases, some preparation needs to be done on a system before data can be copied nondisruptively. Examples include an upgrade of HBA device drivers and replacement of multipathing software. These preparation steps often require downtime because a reboot is needed to activate the installed code. After the preparations are completed, the data can be moved or copied nondisruptively with little or no performance impact.
- Offline migrations: The data must be in a known consistent state before they can be copied or moved. The data must also remain unchanged for the duration of the migration process. Depending on the amount you are migrating, data can remain unavailable for an extended time, from several hours to days.

2.1.1 Host-Based migration

Host-Based migrations use functions provided by the following tools:

- Host operating system (OS) such as Logical Volume Manager (LVM for UNIX and Linux, LDM for Windows)
- Add-on software such as IBM TDMF or Veritas Volume Manager (VxVM)
- Other native volume management tools

Volume management software

Volume management software provides a simple and robust set of tools for selective copying, mirroring, and migration of data. When you perform data migration on critical production systems, you can use volume management software to tune data copying rate to minimize the performance impact.

AIX, Solaris, HP-UX, Linux, Windows, and IBM z/OS are equipped with volume management software that manages disk storage devices. You can use this software to configure and manage volumes, file systems, paging, and swap spaces.
For a detailed description about how to perform these types of migrations, see Chapter 8, “Using mirroring techniques” on page 265.

**File copy**

If the data you are migrating is a group of individual files and no volume management software is available, use a file-level migration technique. This technique uses native OS or third-party utilities and commands that support the file copy feature. Using copy commands and utilities is the simplest form of data migration between two file systems.

In UNIX, you can use the following commands:

- `cp`
- `cpio`
- `tar`
- `pax`
- `dump`
- `backup` and `restore`
- `rsync`

In Windows, you can use the following commands:

- `cp`
- `scopy`
- `xcopy`
- `robocopy`
- `Windows Explorer` drag and drop function

File copy has the following advantages:

- Is available on every OS type that can be attached to IBM disk subsystems
- Can be used to copy between file systems with different sizes, so it can be used for file system and volume consolidation

File copy has the following disadvantages:

- Works only with data organized in file systems format
- To preserve data consistency, all writes to a file must be stopped for the duration of the copy process
- Some copy commands cannot preserve advanced metadata, such as access control lists or permissions

**Important:** If your storage systems are attached through multiple paths, verify that the multipath drivers for the old and new storage can coexist. If they cannot, revert the host to a single path configuration and remove the incompatible driver before attaching the new storage system.

**Raw devices copy**

Data on raw volumes can be read and written directly from and to a volume using tools such as `dd` (a UNIX utility). Such tools are typically used for copying contents of a raw volume from one physical device to another. Take care when an individual volume has external dependencies. An example is a journal file system on one volume that has a corresponding journal on another volume. In this case, you must copy the content of both volumes at the same time. In addition, the configuration of the file systems must be adjusted to reflect the name of the new volumes.
Raw device copy commands such as `dd` provide a powerful interface for copying data between volumes. However, you must ensure that target volume has sufficient space. You must also update all configuration information to reflect the new volume name as in the example.

The raw device copying method is an offline migration method. Applications accessing data on raw logical volumes must remain offline for the duration of the data copying process. The tools do not prevent you from reading data from a volume being used by an application. This might result in inconsistent data on the target volume.

### 2.1.2 Array-Based migrations

An array-based migration uses IBM Copy Services functions to transfer data from one storage subsystem to another. You can use functions provided by Metro Mirror, Global Mirror, and Global Copy to replicate data between compatible storage arrays. This method is addressed in more detail in Chapter 4, “Data migration using IBM Remote Mirror and Copy” on page 41, and *DS8000 Copy Services for IBM System z*, SG24-6787.

### 2.1.3 Appliance-Based migrations

Appliance-Based data migrations use a virtualization device between a server and a disk storage subsystem. This virtualization device acts as a mediator for block level or file level protocols. The virtualization device allows you to migrate data from one physical device to another without any impact on applications and users.

The appliance-based migrations addressed in this book are based on the SAN Volume Controller for block level and f5 ARX-VE for file level storage.

#### SAN Volume Controller-Based migrations

IBM SAN Volume Controller can be used to migrate data between any supported back-end disk storage platforms. To connect to SAN Volume Controller-Managed storage, install and configure the compatible device driver on a server. After a server is connected to SAN Volume Controller managed storage, all data migration activities are done by SAN Volume Controller. There is no interaction with the server software.

> **Important:** SAN Volume Controller-Based migrations can be used only for fixed block type storage type of open systems. CKD storage used by IBM System z® is not supported by SAN Volume Controller.

### 2.1.4 Migrating using backup and restore

Standard UNIX commands can be used to migrate data. The following variations can be used:

- Back up and restore
- Dump and restore
- Other commands

The backup and restore options allow for consolidation because the tools are aware of the data structures they handle.

These methods are unusual in that they do not require the source and target storage systems to be connected to the host at the same time.
Back up and restore
In this variation, you back data up to an external device, such as tape, and restore it to the target location. This method is slow because an extra copy step is introduced and every file or block of data is transferred twice:

- From source to a temporary location
- From the temporary location to the target.

However, if you must remove the old storage system before installing the new one, you must use an external storage device.

Migrating data using backup and restore generally have the most impact on system usage. This process requires that applications and in certain cases file systems be in quiescent states to ensure data consistency.

Dump and restore
These commands are similar to the Backup and Restore command. You can find them on almost all forms of UNIX. This method in most cases also requires an intermediate device.

Other commands
You can find other commands on UNIX systems for backing up data. Again, these commands require that you create an intermediate backup image of an object before restoring to a target location.

You might not be able to use the volume management methods of migrating data in the following cases:

- For databases that use their own storage administration software for managing raw volumes
- Specialized applications that use raw volumes

In some cases, applications use volume names or serial numbers to generate license keys, effectively becoming location dependent. These applications might not tolerate data migration done with tools other than the tools supplied with the application itself such as data export/import tools.

All open systems platforms and many applications provide native backup and restore capabilities. They might not be sophisticated, but they are often suitable in smaller environments. In large data centers, it is customary to have a common backup solution across all systems. Either solution can be used for data migration.

The most common software packages that provide this function are:

- IBM Tivoli Storage Manager
- Legato Networker
- BrightStor ARCserve
- Symantec NetBackup

2.1.5 Data migration in System z environments
This section highlights methods for migrating data from existing disk storage systems onto the DS8000, specifically in a System z environment. Some are similar to the techniques previously explained. The following topics are addressed:

- Data migration based on physical volume migration
- Data migration based on logical data set migration
Combination of physical and logical data migration

Copy Services-based migration

Data migration based on physical volume migration
This method uses physical full volume migration, which requires the same device geometry on the source and target volumes. The device geometry is defined by the track capacity and the number of tracks per cylinder. Usually this is not an issue because over time the device geometry of the IBM 3390 volume has become standardized. For organizations still using other device geometry (for example, 3380), consider a device geometry conversion, if possible. Conversion requires moving the data on a logical level (data set level), which allows a reblocking during the migration from 3380 to 3390.

Physical full volume migration is possible with the following tools:

- Software-based:
  - DFSMSdss
  - TDMF
  - FDRPAS
  - z/OS Global Mirror (XRC)

  **Note:** XRC requires a license on the DS8000.

- Storage-based:
  - IBM Rational® Method Composer functions

Data migration based on logical data set migration
Data logical migration is a migration by data set that maintains catalog entries according to the data movement between volumes. It is not a volume-based migration. Data logical migration is the cleanest way to migrate data, and allows device conversion from, for example, 3380 to 3390. It also transparently supports multivolume data sets. Logical data migration is a software-only approach, and does not rely on volume characteristics or on-device geometries.

The following software products and components can be used for data logical migrations:

- DFSMS allocation management.
- Softek zDMF.
- System utilities such as:
  - IDCAMS using the REPRO and EXPORT/IMPORT commands.
  - IEBCOPY for Partitioned Data Sets (PDS) or Partitioned Data Sets Extended (PDSE).
  - ICEGENER as part of DFSORT, which can handle sequential data but not VSAM data sets.
  - IEBGENER, which has the same restrictions as ICEGENER.

- Database utilities for data that are managed by certain database managers, such as IBM DB2® or IBM IMS™. IBM CICS® as a transaction manager usually uses VSAM data sets.

Combination of physical and logical data migration
The following approach combines physical and logical data migration:

1. Perform a physical full volume copy to a larger capacity volume when both volumes have the same device geometry. They must have the same track size and the same number of tracks per cylinder.
2. Use COPYVOLID to keep the original volume label, and maintain catalog management. You can still locate the data on the target volume through a standard catalog search.

3. Adjust the VTOC of the target volume to make the larger volume size visible to the system. Use the ICKDSF REFORMAT command REFVTPC to refresh the VTOC, or use the command EXTVTOC to expand the VTOC.

**Tip:** Issuing the EXTVTOC command requires you to delete and rebuild the VTOC index using EXTINDEX in the REFORMAT command

4. Perform the logical data set copy operation to the larger volumes. This operation allows you to use either DFSMSdss logical copy operations or the system-managed data approach.

When there are no more data moves because the remaining data sets are in continual use, schedule downtime to move them. You might have to run DFSMSdss jobs from a system that has no active allocations on the volumes that need to be emptied.

### 2.1.6 Summary

Each method of data migration has strengths and limitations. Table 2-1 lists the pros and cons of the suggested products and techniques covered in this book. It is not, however, a comprehensive list of every technology and technique.

*Table 2-1  An overview of migration techniques*

<table>
<thead>
<tr>
<th>Migration technique</th>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
</table>
| Host-based          | - Generally lowest initial implementation costs  
|                     | - Uses existing hosts and IP network  
|                     | - LVM and LDM are parts of the operating system  
|                     | - Storage-vendor-agnostic  
|                     | - Uses existing operating system skills  
|                     | - Migration can happen online during peak hours  
|                     | - Consumes host resources  
|                     | - Operating system specific  
|                     | - Management can become complex and time consuming  
|                     | - No central management console  
|                     | - Might cause an initial outage to install the utility or software  
| Network-based replication | - Supports heterogeneous environments (servers and storage)  
| - Softek TDMF (IP) | - Single point of management for replication services  
|                     | - Does not consume host resources for replication  
|                     | - Higher initial cost due to hardware and replication software  
|                     | - Requires proprietary hardware  
|                     | - Might require implementation of storage networks  
| Array (storage)-based (specifically DS8000 Copy Services) | - High performance  
|                     | - Operating system independent  
|                     | - Does not consume host resources for migration  
|                     | - High cost  
|                     | - Requires knowledge specific to the type of storage platform used for the migration  
|                     | - Complex management  
|                     | - Might require a quiesce of the database  

Chapter 2. Migration techniques and processes  13
### 2.2 Key decision factors

During the planning, you need to determine what hardware or software is needed to successfully perform the migration. Here are key factors to consider and a brief description about how the products highlighted in this book address that specific factor.

#### 2.2.1 Migration performance

One consideration is *migration performance*, which is how quickly data is copied from the source to the target. You need to balance performance against network bandwidth and system resources. For example, data that is copied at a high rate but consumes too many system resources can severely affect production application or system performance. If data is copied too slowly, the migration process can take a significant amount of time, potentially prolonging downtime of an application.

Various tools have the following capabilities to pause, pace, or throttle the migration speed:

- **Appliance-based (SAN Volume Controller)**
  - Migration can happen online during peak hours
  - Supports heterogeneous environments (servers and storage)
  - Single point of management for migration

- **Tape backup/restore-based (Legato, Tivoli Storage Manager, Net-backup, and so on)**
  - Does not require additional special tools, software, and hardware
  - Does not require additional skills or training

#### 2.2.2 Migration backout scenarios, protecting source and target volumes

*Data migration backout scenarios* allow you to roll back the migration. If something goes wrong, you can terminate the migration, and restart or continue application processing on the original data set. With tools such as LVM, you can use a function that moves the data from the source to the target. This approach makes migration backout more complex because it requires all the data to be moved back to the source. Another approach is to mirror and...
synchronize data between the source and the target. This approach allows you to choose which copy of data to use at the time you disassociate the target from the source. The state of the source volume remains preserved and the volume can be used independently from the target volume if anything goes wrong during the migration.

The available tools have the following advantages and disadvantages:

- TDMF allows you to preserve the state of the source volume. If an error occurs during migration, TDMF switches back automatically to the source volumes.
- Host-based utilities such as LVM for UNIX and LDM for Windows can migrate data while applications are up and running. Choose a Volume Manager mirroring technique that allows you to break the mirrored copy of a volume into two or more physical copies. Each physical copy can then be activated, mounted, and used independently. If a copy becomes unusable, it is more practical to revert to another copy. Activating and mounting file systems on an independent copy of a volume is the least time consuming way to back out.
- With SAN Volume Controller, the source volume is first brought under the control of the SAN Volume Controller as an image mode device. The data remains in its original form on the source storage platform. If the host does not recognize the volume presented through the virtualization layer, this step can be undone. The server can then be reconnected to the source volume directly. After the source volume is converted from image to managed mode, the source volume can no longer be accessed directly. You must use the SAN Volume Controller virtualization layer. Converting the volume from image mode to managed mode using SAN Volume Controller is similar to moving data using LVM. When the data movement is done, the source volume can no longer be used. There are SAN Volume Controller tools similar to LVM mirroring where an extra copy of a volume can be created and used for backout. These tools and techniques are addressed in more detail in Chapter 7, “SAN Volume Controller-based migration” on page 175.
- IBM Copy Services using Rational Method Composer allows the migration to be terminated and restarted from the beginning, or paused and restarted from the last write to the target disk.

### 2.2.3 Understand target storage requirements

Before starting the data migration to a target storage platform, you need to plan for optimal volume allocation at the target. For environments with no storage virtualization capabilities, data migration presents an opportunity for storage optimization. If planned properly, you can achieve the following improvements in volume allocation:

- Performance optimization
- Resiliency
- Migration to another storage tier
- Reclamation of unused capacity

Before the data migration takes place, create a set of target volumes. Thorough planning for the target volume allocation is essential for storage optimization. Storage optimization is a complex task and the optimization goals can be contradictory. You might need to prioritize based on your company standards and available tools and techniques available for the data migration. Any mistakes made at this stage of the data migration process will be difficult to correct in an environment with non-virtualized storage.

A virtualized storage environment can have nondisruptive data movement, so storage optimization that can be performed without affecting applications and users. In a non-virtualized storage environment, however, the data migration is a disruptive process. Often the only opportunity for optimization of non-virtualized storage is when the data is migrated.
Understanding the architecture of a target disk platform is important during the planning phase. If the architecture is not properly understood, you might create configurations with suboptimal performance, availability, and scalability characteristics. Understand your key applications disk access patterns so you can optimize performance and availability characteristics of the most critical applications. Storage tiering is a common technique used for workload prioritization. However, even after an application is identified for a specific storage tier, there are other allocation aspects that need to be considered:

- Improving the resiliency of a database server
- Avoid having database volumes and transaction logs on the same RAID group. Careful placement reduces the risk of losing data as a result of a RAID group failure.

Architecture of various disk platforms, allocation techniques, connectivity, storage tiering, and alignment with applications are complex tasks. For more information about architectural considerations for specific disk subsystems, see 2.4, "Preparing DS8000 for data migration" on page 27.

Migration tools have different capabilities for migrating data to volumes with differing capacities. Tools such as UNIX LVM-based tools are the most flexible when it comes to dealing with the differences in volume capacities. Windows LDM migrations require the target volume to be the same size or larger.

IBM Copy Services-Based migrations in most cases require the target volume to be of the same capacity as the source. Generally, work with matching capacity volumes. You might need to reverse the direction of data copying, which is not allowed from a volume of greater capacity to a lesser one. After the migration is completed, you can freely increase the capacity of a volume if required. Use operating system tools to recognize the increased capacity of an underlined volume, and to expand boundaries of a file system on that volume.

SVC-Based migration from image mode to managed mode virtual disk cannot involve any volume capacity changes. After the migration is finished, the target volume capacity can be dynamically expanded. This process is similar to SVC-Based data migrations using Copy Services or volume mirroring techniques.

**Important:** Be careful that the data capacity on the source volume is not greater than the free space on the target volume.

### 2.2.4 Different source and target hardware

Another common migration situation is *unlike source and target storage hardware*. Although host-based products allow migration to unlike storage devices, ensure the DS8000 minimum level requirements are met for array-based migrations. The different tools have the following considerations:

- TDMF is hardware-independent, allowing nondisruptive migration in multivendor environments.
- SAN Volume Controller provides a virtualization level that isolates the back-end storage type from the host. It therefore allows any supported storage servers data to be migrated to any other type of supported storage platform.
Host-based LVM/LDM tools allow for unlike source and target platforms, but might be limited in which device drivers can coexist on the host operating system. Research driver compatibility with the system administrators for each type of platform and operating system type.

Tape-based products run on just about any type of platform and operating system. Check with your backup/recovery administrator for information.

2.2.5 Application downtime

One of the primary reasons that data migrations occur during off-hours is to avoid application downtime during peak periods. Depending on the type of data and applications you are migrating, you might have only a narrow downtime window. For example, having the HR system is offline for 24 hours over a weekend would have less business impact than if the e-commerce system were offline for the same period. Some systems are so critical that a few hours or even minutes of downtime, even during off-hours, is unacceptable. If you are performing a significant upgrade of such a system, using offline data migration tools and techniques is not feasible.

The different tools have the following capabilities:

- TDMF allows for nondisruptive data migration, so applications can stay online and continue to process data throughout the migration process. Some non-IBM products cannot be moved online. For more information, see the TDMF Installation and Reference Guide at:
  

- SAN Volume Controller requires an initial reboot to load the device drivers on the host. It also requires the application to be brought offline while the source volume is discovered. You can minimize this outage by performing these steps in parallel. The rest of the migration is done online.

- LVM/LDM does not require any outage and in certain cases can be done online. The only step of the emigration process that requires an outage is the HBA and multipathing device driver installation. An outage is not necessary if the current version of the HBA and multipathing device driver supports the source and the target disks subsystems. In most cases, however, an upgrade of HBA device driver is required even if data migration is done between disk subsystem of the same type. Different versions of disk subsystem microcode levels often require different versions of HBA BIOS and device driver levels to ensure compatibility and supportability.

- IBM Copy Services allow for seamless data copying between the source and the target. The data is instantaneously available after a synchronous Copy Services relationship connection is stopped. If a relationship uses asynchronous connection, there will be some delay between last write to the source volume and when the data is fully synchronized. When the data copying is finished, the server must be disconnected from the source and reconnected to the target, applications shutdown, and file systems demounted. This process might require a reboot depending on the levels of the HBA and multipathing device drivers. Reboot your server after reconnecting to the target disk subsystem to verify that the migration is done properly and all file systems are online.
2.2.6 Key decision factors summary

Table 2-2 summarizes the factors to consider when migrating data, and what tools address those concerns.

<table>
<thead>
<tr>
<th>Key factors</th>
<th>Description</th>
<th>Capability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance</td>
<td>How quickly can data be copied from the source to the target, balanced against system resources</td>
<td>▶ TDMF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>▶ SAN Volume Controller</td>
</tr>
<tr>
<td></td>
<td></td>
<td>▶ IBM Copy Services</td>
</tr>
<tr>
<td></td>
<td></td>
<td>▶ LVM</td>
</tr>
<tr>
<td>Primary volume/Source data protection</td>
<td>If something goes wrong, the migration can be terminated and application processing restarted or continued on the source data or device</td>
<td>▶ TDMF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>▶ SAN Volume Controller with limitations</td>
</tr>
<tr>
<td></td>
<td></td>
<td>▶ LVM / LDM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>▶ Tape based</td>
</tr>
<tr>
<td></td>
<td></td>
<td>▶ IBM Copy Services</td>
</tr>
<tr>
<td>Implement tiered storage</td>
<td>Moving data to a different array or to different storage media for cost performance without disruption</td>
<td>▶ TDMF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>▶ SAN Volume Controller</td>
</tr>
<tr>
<td></td>
<td></td>
<td>▶ LVM / LDM</td>
</tr>
<tr>
<td>Multi-vendor environments</td>
<td>Hardware from several vendors is in use, which can result in data migration between different vendor disk platforms</td>
<td>▶ TDMF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>▶ SAN Volume Controller</td>
</tr>
<tr>
<td></td>
<td></td>
<td>▶ LVM/LDM with possible restrictions</td>
</tr>
<tr>
<td></td>
<td></td>
<td>▶ Tape-based</td>
</tr>
<tr>
<td>Application downtime</td>
<td>Applications have different levels of business criticality and therefore have varying degrees of acceptable downtime</td>
<td>All with limits</td>
</tr>
<tr>
<td></td>
<td></td>
<td>▶ TDMF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>▶ SAN Volume Controller</td>
</tr>
<tr>
<td></td>
<td></td>
<td>▶ LVM / LDM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>▶ IBM Copy Services</td>
</tr>
</tbody>
</table>

Note: Features and functions can vary by operating system.

2.3 Data migration process

Data migration is typically a complex process. To minimize the impact of downtime, reduce the possibility of data loss, and control costs, employ a consistent, reliable, and repeatable migration methodology.

The migration process includes the following phases:

▶ Planning phase
▶ Migration phase
▶ Validation phase
A typical migration process is shown in Figure 2-1.

### Figure 2-1  A suggested three-phase migration process

#### 2.3.1 Planning phase

A successful data migration always requires substantial time spent on evaluation and planning. Adequate planning is the critical success factor in any migration project.

The higher the complexity of an environment and the more critical the data, the more critical migration planning becomes. Careful migration planning identifies potential issues, allowing you to avoid or mitigate them. Migration planning also identifies which data to migrate first, which applications must be taken offline, and the internal and external colleagues you need to communicate with.

**Important:** Planning is the critical success factor in a migration project.

The planning process must cover the following considerations:

- Determine the requirements for the migration
- Assess the current storage environment
- Identify requirements for the future state of the environment
- Create a migration plan
- Document architectural decisions for specific hardware and software components
- Develop migration procedures
- Test plans

Successful migration planning involves more than just the IT staff. The business owners of the applications and data being migrated must also be included. The owners are the best resource for determining how important an application or set of data is to the business.

Coordinate the migration with the application owners because uncoordinated tasks can cause problems during the migration. Do not, for example, plan a migration of the financial system...
on the same weekend that the finance department is finalizing quarterly numbers. Keep all affected application owners involved and informed throughout the migration process.

**Assessment of the environment**

Involve the application owner in the early stages of the migration planning process so all affected applications are considered. Many applications have dependencies on other applications, some of which might not be obvious. These dependencies can be at any layer of the application stack. For example, a storage device that requires a physical move to another location affects application A. Because of the resulting bandwidth restriction, the application server must also be moved to the new location. However, Application B must be on the same IP subnet as application A, so it must be migrated as well.

To uncover and understand the implications of such dependencies, carefully assess the complete environment using the following steps:

1. Identify all applications that are directly affected by the planned migration.
2. Find application dependencies throughout the entire application stack, all the way down to the hardware layer.
3. If any additional cross-application dependencies are identified, expand the migration scope to include the newly identified application.
4. Repeat the previous steps until no additional cross-application dependencies are uncovered.
5. Identify and document the requirements for a downtime of all the applications in scope.
6. Evaluate the requirements against any restrictions for application downtime.

Additional constraints might be identified during the assessment process, including the need for extra capacity, connectivity, network bandwidth, and so on. Discovering these additional requirements early is vital because it gives the project team more time to address them or develop alternatives.

**Test and develop migration scenarios**

Usually, more than one set of tools and techniques can be used in a data migration. You need to evaluate the available tools based on your type of migration and the outcome of the assessment process. To test these migration scenarios, build a test environment that duplicates as near as possible the environment on which the migration will take place. If you cannot construct such an environment, IBM has a service offering that allows you to use IBM lab equipment. The migration process can then be tested.

When the migration scenarios are developed in the test environment, plan to be able to backout the migration so you can effectively reset the test environment. When all scenarios have been developed, tested them extensively until you are confident that the scenarios will work during the migration of live data.

You might also want to develop automation scripts. The following examples demonstrate situations where automation scripts are useful:

- When the time frame of the migration is weeks or months. Changes to the production environment, like storage expansion, might happen during the migration, and these additional volumes must be taken into account.
- When the scenarios are so complex that operators need guidance. These instructions can be provided with a script. For example, in a migration where the scenarios are issued in a defined sequence, the operator can be guided step by step using a script.
When reliable status verifications are difficult to get, especially in a situation where many volumes must be migrated. A script automating the status check can reduce errors, particularly when the volume addresses are not in a contiguous order.

Budget sufficient time and attention for this important phase. The better you test the scenarios and provide automation, the smoother the live migration will go. The team running the migration must be familiar with every step of the development process, and must feel confident with the proposed timelines. Strive to minimize the risk of unplanned outages or other complications during the migration.

Thorough development and testing of the migration process can reduce the potential impact of the migration and increasing success rate of individual migrations.

**Migration plan**

As part of the planning and preparation, create a high-level migration plan (Table 2-3) and communicate the plan to all stakeholders.

Employ checkpoints to ensure that the migration is proceeding correctly. It can be overwhelming, for example, to consider migrating several servers in one large move. Instead, break down the migration into smaller activities to make it more manageable. Among other benefits, checkpoints simplify any needed backouts. Therefore, create an overall comprehensive plan and then break individual tasks into subplans.

The plan also allows you to track schedule commitments while completing the migration. Always allocate extra time for tasks, generally 15 - 20% more time than would be required for the best case migration scenario. This gives you time to resolve unexpected issues. The better you plan, the fewer unexpected or unforeseen issues will be encountered with both resource (team members) commitments and technology.

**Table 2-3  Sample high-level migration plan**

<table>
<thead>
<tr>
<th>Action Item</th>
<th>Assigned to</th>
<th>Status</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Establish a migration management team.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Gather availability and production schedules.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Document Change Control procedures so they can be incorporated into the migration procedures and plans.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Document the timeline for activities for both hardware changes and the data migration.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5. Announce the migration early in the cycle. Adhere to the Change Management process to determine how much advanced notice needs to be given.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6. Gather information about the storage server environment and applications (lists and drawings).</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7. Inform the security and compliance groups about the migration.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Data Migration to IBM Disk Storage Systems

Detailed information list for the migration plan

This section addresses the migration plan action items in more detail.

1. Establish a migration management team and create a technical migration team.

   Some environments have team members who have several roles and responsibilities. Team members might include, but are not limited to, the following roles:
   
   - Project manager
   - Client (account) manager
   - DBA/Application owners
   - System administrator
   - Network administrator
   - Security administrator
   - Firewall administrator
   - Disk storage administrator
   - Backup/Recovery administrator
   - SAN fabric administrator
   - Hardware CE
   - Floor planner
   - Cable vendor
   - Disaster/Recovery administrator
   - IT Architect

2. Gather availability and production schedules.

   Determine what outage windows are available or need to be scheduled. Some activities call for multiple outages. Planning outage times will help the overall migration run smoothly.

   Application availability can be limited for the following reasons:
   
   - Month/end/end quarterly processes
   - IBM FlashCopy® or Metro/Global mirror/copy running processes and their time restrictions
   - Database/application refreshes

3. Document Change Control procedures so they can be incorporated into the migration procedures and plans.

   Educate all team members on using the change control tool. Successful coordination allows tasks owned by individual team members to be run in parallel or in a timely fashion. Some change procedures require lead times of seven days or more before they can be completed. Any emergency changes can also be communicated to the team.
Following the Change Control process helps avoid missing Service Level Agreement (SLA) and contractual obligations.

4. Document the timeline for activities for both hardware changes and the data migration.

   Develop a realistic timeline with built-in buffer times to create realistic client and team expectations. It can also show if your progress in the overall migration is on schedule, allowing you to more adjust and re-evaluate the plan as needed.

5. Announce the migration early in the cycle. Use the Change Management process to determine how much advanced notice is needed.

   Announcing early allows you enough time to adjust the plan if something is discovered that might cause a potential slowdown. It is critical to warn all affected parties before touching and moving data. Keep application owners informed by publicizing the migration plan and giving them a chance to identify flaws.

6. Gather information about the storage server environment and applications (lists and drawings).

   You must understand the following design requirements thoroughly:
   – Migration and replication requirements,
   – Time schedule
   – Vendors involved
   – Configuration of the hardware.

   When sizing data migrations, there are many key items to consider:
   – The number of servers
   – The operating system levels
   – The amount of storage
   – The volume managers
   – Types of databases and applications
   – Network speeds
   – Server clusters

   When creating the time, create estimates for planning (typically 25% of the total), installation and setup time, data copy time, and production cutover.

   Your specific requirements help determine the best technology to use for your migration.

   Make a list of all the components involved in the migration like the example shown in Table 2-4.

Table 2-4  Design requirements

<table>
<thead>
<tr>
<th>Action item</th>
<th>Server environment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Server manufacturer (Compaq, Dell, IBM, HP, Sun)</td>
</tr>
<tr>
<td></td>
<td>Number of processors</td>
</tr>
<tr>
<td></td>
<td>Number of logical partitions (LPARs) or domains</td>
</tr>
<tr>
<td></td>
<td>Type of file system (UFS, VxFS, HFS, JFS, JFS2 (inline or outline), NFS, NTFS, FAT, FAT32</td>
</tr>
<tr>
<td></td>
<td>Operating system (OS) version (AIX 5.1, zOS 1.4, IBM System i®)</td>
</tr>
<tr>
<td></td>
<td>OS addressing (31-bit, 32-bit, 64-bit)</td>
</tr>
<tr>
<td></td>
<td>Databases to be moved (DB2, IBM Informix®, Oracle, SQL, Sybase)</td>
</tr>
<tr>
<td></td>
<td>Database version</td>
</tr>
<tr>
<td></td>
<td>Database size</td>
</tr>
</tbody>
</table>

Chapter 2. Migration techniques and processes  23
7. Identify target storage requirements.
   Work with your storage administrator, to understand and outline the new storage configuration and layout. Identify specific requirements for volume RAID type, size, and disk and spindle isolation versus spread ahead of time. For more information, see 2.4, “Preparing DS8000 for data migration” on page 27.

8. Inform the security and compliance groups about the migration.

9. Schedule a pre-migration rehearsal that includes all the members on the migration team and a representative data sampling.
   A rehearsal allows you to uncover any flaws in your plan. Ideally, test a complete migration cycle using test equipment.

10. Establish a Migration Status call-in process.
    Having a regular phone meeting during the migration helps keep the migration on schedule and avoid problems.

11. Use a Migration Planning checklist to make sure that all pre-migration planning steps are executed.

Table 2-5 illustrates an example checklist for migration planning.

<table>
<thead>
<tr>
<th>Action item</th>
<th>Migration and validation methodology checklist</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Structure the migration architecture to match the production requirements.</td>
</tr>
<tr>
<td></td>
<td>Use checklists to ensure that operating patches and software are at the correct levels.</td>
</tr>
</tbody>
</table>
Migration phase
During the migration phase, communicate your plans and obtain, install, and configure hardware, software, automation scripts, and tools needed to perform the actual data migration. Run a pre-migration data validation test in addition to post-migration validation testing. For more information about validation, see 2.3.2, “Validation phase” on page 27. These tests confirm that the data is in the same state after the migration as it was before. Test your plan on a test or development (non-production) environment if possible.

The most important part of this stage is the actual migration itself. As outlined, proper methodology can simplify this process by:

- Enhancing the speed of migration
- Minimizing or eliminating application downtime
- Allowing migration during regular business hours

Table 2-6 illustrates a high-level sample plan only. Customize it for your specific environment. You can use it as a model, or write your own plan. Part 2, “Migration scenarios” on page 39 contains several sample plans for host-based, network-based, array-based, and appliance-based migration plans. See the sample plan that best matches your migration to become familiar with the concepts and principles. The details need to be worked out with the migration team members. This example plan illustrates what steps might be taken to migrate Windows SAN volumes to a DS8000 using SAN Volume Controller.

Table 2-6  High-level test project plan for a single host server

<table>
<thead>
<tr>
<th>Location</th>
<th>Activity</th>
<th>Owner</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Validate Change Record is in place and is approved</td>
<td>Storage Admin</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Verify the WWPN numbers of the HBAs with the System Admins</td>
<td>Storage Admin and System Admin</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Update SDD drivers</td>
<td>System Admin</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reboot host server</td>
<td>System Admin</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Validate ESS LUNs can still be accessed</td>
<td>System Admin</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Power down host server</td>
<td>System Admin</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Location</td>
<td>Activity</td>
<td>Owner</td>
<td>Comment</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-----------------------------------------------------------------</td>
<td>------------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td></td>
<td>Contact Storage Team when finished</td>
<td>System Admin</td>
<td></td>
</tr>
<tr>
<td>ESS unit</td>
<td>Locate LUNs on ESS unit for this host server</td>
<td>Storage Team</td>
<td></td>
</tr>
<tr>
<td>ESS unit</td>
<td>Unassign ESS LUNs from host server</td>
<td>Storage Team</td>
<td></td>
</tr>
<tr>
<td>ESS unit</td>
<td>Assign ESS LUNs to the SAN Volume Controller</td>
<td>Storage Team</td>
<td></td>
</tr>
<tr>
<td>SAN Volume Controller cluster</td>
<td>Create host definition</td>
<td>Storage Team</td>
<td></td>
</tr>
<tr>
<td>SAN Volume Controller cluster</td>
<td>Discover Mdisk</td>
<td>Storage Team</td>
<td></td>
</tr>
<tr>
<td>SAN Volume Controller cluster</td>
<td>Change Mdisk names</td>
<td>Storage Team</td>
<td></td>
</tr>
<tr>
<td>SAN Volume Controller cluster</td>
<td>Create Image Mode/Unmanaged Vdisks in Mdiskgroup IMAGEGRP</td>
<td>Storage Team</td>
<td></td>
</tr>
<tr>
<td>SAN Volume Controller cluster</td>
<td>Map Vdisks to host</td>
<td>Storage Team</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Contact System Admin team when completed</td>
<td>Storage Team</td>
<td></td>
</tr>
<tr>
<td>Host server</td>
<td>Power up host server</td>
<td>System Admin</td>
<td></td>
</tr>
<tr>
<td>Host server</td>
<td>Validate Vdisks can be accessed by host server</td>
<td>System Admin</td>
<td></td>
</tr>
<tr>
<td>Host server</td>
<td>Start applications</td>
<td>System Admin</td>
<td></td>
</tr>
<tr>
<td>Host server</td>
<td>Turn over to customer</td>
<td>System Admin</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Contact System Admin Team when completed</td>
<td>System Admin</td>
<td></td>
</tr>
<tr>
<td>SAN Volume Controller cluster</td>
<td>Migrate ESS Vdisks to DS8300 Mdiskgroup</td>
<td>Storage Team</td>
<td></td>
</tr>
<tr>
<td>SAN Volume Controller cluster</td>
<td>Rename ESS Vdisks</td>
<td>Storage Team</td>
<td></td>
</tr>
<tr>
<td>SAN Volume Controller cluster</td>
<td>Remove ESS Mdisk from IMAGEGRP Mdiskgroup</td>
<td>Storage Team</td>
<td></td>
</tr>
<tr>
<td>ESS unit</td>
<td>Unassign ESS LUNs from SAN Volume Controller</td>
<td>Storage Team</td>
<td></td>
</tr>
</tbody>
</table>
2.3.2 Validation phase

After migrating data, ensure that the target system contains a complete and accurate copy of the source. You also need to make sure that the migrated data is still usable by applications.

*Note:* It is important to validate that you have the same data and functions of the application after the migration. Make sure that the application runs with the new LUNs, that performance is still adequate, and that operations and scripts work with the new system.

After you complete the migration, compile migration statistics and prepare a report to highlight what worked, what did not work, and any lessons learned. Share this report with all members of the migration team.

These reports are critical in building a repeatable and consistent process by building on what worked, and fixing or changing what did not. Also, documenting the migration process can help you train your staff, and simplify or streamline the next migration you do, reducing both expense and risk.

Make sure during the validation phase that the backout scenarios can still be applied. If there are problems with the data or the infrastructure after the migration, you can fix the problem or revert to the original state.

Also, keep the data at the original site available and usable. Maintaining this data allows you to discard the migrated data and restart production at the original site if something goes wrong with the migration. Another migration can then be run after you fix the problems without affecting your applications.

### 2.4 Preparing DS8000 for data migration

When planning a migration to the DS8000, review the overall concepts and architecture of the storage subsystem. This review helps you to prepare the logical system configuration for the databases or data files you are migrating.

Make sure that you understand and document the following items:

1. The logical layout of the Arrays and LUNs (understanding array characteristics with workloads and data placement on arrays and LUNs)
   a. The size of the arrays
   b. The number of arrays
   c. The type of arrays (6+P) versus (7+P)
   d. The array locations in relationship to the owning DA Pair
   e. Logical LUN (Volume) sizes versus physical disk sizes

<table>
<thead>
<tr>
<th>Location</th>
<th>Activity</th>
<th>Owner</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESS unit</td>
<td>Remove host definition from ESS unit</td>
<td>Storage Team</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Remove the ESS to host zone</td>
<td>Storage Team</td>
<td></td>
</tr>
<tr>
<td>SAN Volume Controller</td>
<td>Validate Mdisks no longer defined on SAN Volume Controller</td>
<td>Storage Team</td>
<td></td>
</tr>
</tbody>
</table>
f. The number and size of DS8000 LUNs you create to match the source Storage layout
g. RAID levels
h. Host striping

2. The DS8000 microcode level and compatibility to support the host SDD or multipathing driver

2.4.1 Available hardware resources

Place data based on the hardware resources available. Determine how many of the following hardware resources you have:

- DA pairs
- Ranks (arrays, array sites)

**Tip:** You will always see two servers (0 and 1).

- I/O enclosures
- Host adapters
- I/O ports are available on each host adapter

Evenly distribute the data across as many hardware resources as possible. However, you might want to isolate data to specific hardware resources for guaranteed resource dedication to that data I/O. For best performance results, create only one rank per extent pool. Limiting the ranks helps you to map and identify problem resources throughout the lifetime of the data or database. Performance issues might arise later because of the following changes:

- Database growth
- Saturation of certain hardware resources as the database changes
- New data or files/filesystems being created on the same set of hardware resources, changing the performance of all data on those resources

2.4.2 Understanding array characteristics

It is beyond the scope of this book to address the performance characteristics of arrays, LUNs and the logical layout of the DS8000. The following is a basic understanding of these characteristics. Preparing the DS8000 properly can minimize performance impacts on the application databases and file systems both during and after the migration.

**Remember:** The following explanation applies only to the initial creation of LUNs in the array.

Figure 2-2 on page 29 shows an example of how the data workloads on the disk arrays affect one another. The example is a DS8000 array of eight DDMs, in a 7+P format with no spare. The eight physical disks are divided into 5 logical LUNs.

The first LUN (logical-disk1) is formed from strips of sections (numbered 1) along the outer edge of each of the DDMs making up the array. Subsequent LUNs are similarly formed from areas on the DDMs, each sequentially closer to the center.
Chapter 2. Migration techniques and processes

Figure 2-2  How LUNs are created in an array

Tip: This example is only true if there is a one-to-one relationship between rank and extent pool. If an extent pool has multiple ranks, the LUN might span ranks.

LUNs from the same array can be assigned to the same or separate servers on an individual basis. Workload sharing or isolation depends on the LUN to array mapping. An example of two databases or workloads on the same array is shown in Figure 2-3.

Figure 2-3  How databases share an array

A database might consists of two of the LUNs (logical volumes made up of strips 1 and 3) in the array. Another database might consist of the other three logical volumes (logical disks 2, 4, and 5) in the array shown in Figure 2-3. The volumes in the array can even be assigned to two different servers.

Because the disks share physical heads and spindles, I/O contention can result if the two application workloads peak at the same time. Prepare the logical configuration of arrays, ranks, extent pools, and LUNs for the DS8000. You must meet or exceed the I/O throughput parameters of the source storage platform you are migrating from.

Larger arrays always outperform smaller arrays. For example, an RAID-5 (7+P) array outperforms a (6+P) array. Data that needs to perform better would do better on a 7+P array when using RAID-5 and (4x4), rather than (3x3) when using RAID-10 arrays.

Important: The speed of the drives is also a factor to consider. An array made up of 15-K DDMs outperforms one consisting of 10-K DDMs. Speed is an important consideration when moving to a DS8000 with mixed speed DDMs.

The 7+P or 4x4 array outperforms a 6+P or 3x3 array because the array will stripe data across more disks. For example, writing data across a 7+P RAID-5 array will stripe across
eight disks (Figure 2-2 on page 29) rather than seven disks (Figure 2-4). A 4x4 RAID-10 array writes the same data twice striped across four disks rather than three disks.

Figure 2-4   A 6+P Array with a spare

Generally, however, try to balance workload activity evenly across RAID arrays, regardless of the size. The cache mitigates most of the performance differences, but keep in mind this guideline when you are fine-tuning the DS8000 I/O throughput.
IBM service offerings

IBM Data Migration Services, part of IBM Storage and Data Product Services, can provide resources, technology, and expertise. IBM Data Migration Services can help you simplify your data migration process in open systems and mainframe environments.

The services combine the experience of IBM services professionals with the following powerful migration tools:

- IBM System Storage SAN Volume Controller
- Transparent Data Migration Facility (TDMF)
- z/OS Dataset Mobility Facility (zDMF)

IBM can help with project planning and management, and can provide technical assistance for migrating data including the following activities:

- Hardware environment refreshes
- Storage reclamation
- Consolidation
- Data center migrations

Data Migration Services are one part of the IBM Global Services portfolio of Data Center Services. Data Center Services includes Storage Optimization and Integration Services for end to end storage consulting.

The following data migration services from IBM Global Services are currently available:

- IBM Migration Services for data for Open Systems
- IBM Migration Service for System z Data
- IBM Migration Services for Network Attached Storage Systems

In partnership with IBM Global Services, IBM Systems and Technology Group offers services for early adopters of IBM technology and the following migration services:

- IBM XIV Migration Services
- IBM DS8000 Data Migration Services using temporary Licenses for Copy Services

This chapter includes the following sections:

- IBM Global Services offerings
- IBM Systems and Technology Group Lab Services
3.1 IBM Global Services offerings

The following services are provided by IBM Global Services divisions around the world:

- IBM Migration Services for data for Open Systems
- IBM Migration Services for System z data
- IBM Migration Services for Network Attached Storage Systems

3.1.1 IBMMigration Services for data for Open Systems

IBM Migration Services for open systems attached to System Storage disk systems provides services performed by a technical specialist to plan and manage nondisruptive migration. Migration is to IBM System Storage Technologies from servers running the following operating systems:

- IBM AIX
- HP-UX
- Sun Microsystems Solaris
- Microsoft Windows technology

This migration can be accomplished with minimal and often no interruption to service using mirroring. Mirroring is an operating system or software/hardware tool. It allows your data to be replicated to the IBM System Storage disk system or other storage vendor products. In addition, IBM can provide a migration control book that details the activities performed during delivery of the services.

In executing these projects, IBM uses the following technologies and products, among others:

- IBM System Storage SAN Volume Controller
- OS-Specific mirroring
- Global Copy
- Softek Transparent Data Migration Facility (TDMF)

Value proposition

Using IBM Migration Services can help you achieve the following goals:

- Reduce or eliminate downtime and data loss
- Preserve data updates throughout the migration, allowing the process to be interrupted if necessary
- Improve post-migration management (a migration control book is provided that explains the work performed)

Benefits

The implemented solution allows you to realize the following benefits:

- Professional, speedy, and efficient planning and implementation of data migration
- Greater flexibility and improved data migration capabilities
- Skills instruction for members of your staff
- Focus on business-critical activities
- Opportunities to reduce the total cost of your IT infrastructure
Key questions and considerations
Use following questions to determine whether IBM Migration Services are right for you:

- Do you have a requirement to efficiently migrate your data?
- Do you need assistance in the planning and execution of data migration?
- Do you have the available resources and technical expertise within your enterprise to manage your data migration?
- Does your staff have the necessary skills to perform this migration?
- Can you afford to have outages during the process?

Deal size/pricing
Engagement pricing varies depends on the following factors:

- Amount of data being migrated
- Complexity and type of data being migrated
- Migration method and technology
- Travel and living expenses

Learn more
IBM TotalStorage hardware-assisted data migration services are available around the world. For more information, visit the following web address:


3.1.2 IBM Migration Services for System z data

IBM Migration Services for System z data provides professional services performed by technical specialists to help plan and manage the nondisruptive migration of data. Use IBM Migration Services for System z data if you want to migrate data to DS8000 or other IBM disk systems from System z attached disk systems.

In addition to providing support for moving data to IBM System Storage products, IBM can also help you move data among disk systems from other storage manufacturers.

Migration can be accomplished using hardware or software that allows direct access to storage device volumes to be copied to new storage devices. The migration takes place without interruption to data availability. IBM can work with your personnel to plan the data migration activities, and can install the migration software or migration hardware tools in your environment.

At the completion of these services, data can be transferred from your existing 3380/3390 formatted DASD volumes to the TotalStorage disk system.

In executing these projects, IBM uses technologies and products such as:

- Softek Transparent Data Migration Facility for z/OS (TDMF z/OS)
- Softek z/OS Dataset Mobility Facility (zDMF).
- Global Copy

Value proposition
Using IBM Migration Services can help you achieve the following goals:

- Reduce or eliminate downtime
- Preserve data throughout the migration
▸ Allows movement to and from 3990-compatible storage systems without special configuration changes
▸ Allows movement to and from IBM disk systems, and systems from other leading manufacturers
▸ Combine or split subsystems
▸ Move portions of data in an existing storage subsystem
▸ Protect data by preserving data updates throughout the migration, allowing the process to be interrupted if necessary
▸ Improve post-migration management (a migration control book is provided that explains the work performed)

**Benefits**
The implemented solution allows you to realize the following benefits:
▸ Professional, speedy, and efficient planning and implementation of data migration
▸ Greater flexibility and improved data migration capabilities
▸ Skills instruction for members of your staff
▸ Focus on business-critical activities
▸ Opportunities to reduce the total cost of your IT infrastructure

**Key questions and considerations**
Use following questions to determine whether IBM Migration Services are right for you:
▸ Do you have a requirement to efficiently migrate your data?
▸ Do you need assistance in the planning and execution of data migration?
▸ Do you have the available resources and technical expertise within your enterprise to manage your data migration?
▸ Does your staff have the necessary skills to perform this migration?
▸ Can you afford to have outages during the process?
▸ Do you need to migrate individual volumes (not subsystems) so you can migrate multiple volumes at the same time?
▸ Do you need to combine or split subsystems, or move portions of data in an existing data storage?

**Deal size and pricing**
Engagement pricing varies depending on the following factors:
▸ Amount of data being migrated
▸ Complexity and type of data being migrated
▸ Migration method and technology
▸ Travel and living expenses

**Learn more**
IBM TotalStorage hardware-assisted data migration services are available around the world. For more information, visit the following web address:

3.1.3 IBM Migration Services for Network Attached Storage Systems

IBM Migration Services for Network Attached Storage provides migration services to help you deploy IBM network-attached storage products. IBM offers specialists who can help you with planning and managing the entire data migration process. They will address issues related to complexity, disruption, system performance, hosting, availability, and data integrity.

Value Proposition
IBM Migration Services for Network Attached Storage combines proven migration methods and tools with the planning and management experience of highly skilled IBM storage specialists. IBM offers you continuous access to critical data throughout the migration process to mitigate project risk. IBM also uses new storage and data platforms for the best IT and business results. Additionally, this service provides both hardware- and software-based migration options that are customized to your needs.

Benefits
The implemented solution allows you to realize the following benefits:

- Increased value of investment in network-attached storage because of faster migration
- Reduced system downtime
- Reduced risk associated with data transfers

Key Questions and Considerations
Use following questions to determine whether IBM Migration Services are right for you:

- Do you have a requirement to efficiently migrate your data?
- Do you need assistance in the planning and execution of data migration?
- Do you have the available resources and technical expertise within your enterprise to manage your data migration?
- Does your staff have the necessary skills to perform this migration?
- Can you afford to have outages during the process?
- Do you need to migrate individual volumes (not subsystems) so you can migrate multiple volumes at the same time?
- Do you need to combine or split subsystems, or move portions of data in an existing data storage?

Deal size and pricing
Engagement pricing varies depending on the following factors:

- Amount of data being migrated
- Complexity and type of data being migrated
- Migration method and technology
- Travel and living expenses

Learn More
IBM Migration Services for Network Attached Storage (NAS) are available around the world. For more information, visit the following web address:

3.2 IBM Systems and Technology Group Lab Services

IBM Systems and Technology Group (STG) Lab Services can help you optimize your data center and system solutions.

STG Lab Services has the knowledge and skills to support your entire information technology solution. STG Lab Services is focused on the delivery of new technologies and niche offerings. It collaborates with IBM Global Services and IBM Business Partners to provide complete solutions to keep your business competitive.

3.2.1 IBM XIV Migration Services

IBM XIV Migration Services from STG Lab Services can help you achieve an efficient and smooth migration of data to the XIV Storage System with minimal disruption. These services assist you with the planning, implementation, and validation of data migration from existing supported hosts and storage subsystems to the new XIV controller.

XIV Migration Services performs the following tasks:

- Conduct an on-site planning meeting to address optimizing migration and define a sample migration exercise for a test or production environment
- Determine personnel skills and other resources required
- Establish your business requirements including performance, capacity, availability, and identification of specific data to be migrated
- Identify security options and considerations
- Assess your migration readiness
- Schedule migration activities, including start and end dates
- Perform migration activities on supported host systems
- Perform post migration verification on a sample set of data
- Conduct informal skills transfer throughout the migration activities
- Update the installation record with details about LUNs migrated
- Provide basic migration skills instruction for up to three (3) designated technical personnel

Value Proposition

IBM XIV Migration Services from STG Lab Services assists you with data migration to an IBM XIV Storage System quickly and with minimal disruption. Services include planning, implementation, validation of data migration, and skills transfer.

Benefits

Using IBM XIV Migration Services from STG Lab Services provides these benefits:

- Correct migration of data to the XIV Storage System
- Shorter migration schedule
- Expert storage architects
- Reduced risk of disruption to business

Key Questions and Considerations

Use following questions to determine whether IBM XIV Migration Services are right for you:

- Do you have the available skills to effectively migrate data to the IBM XIV Storage System?
If you have the skills internally, would it be a better use of your resources to have IBM provide XIV migration services?

Would you like to increase the XIV migration skills of your staff?

Do you want to have your data migrated to your new IBM XIV Storage System by highly skilled and experienced specialists?

**Deal size and pricing**
Contact an STG Lab Services Opportunity Manager for an estimate of pricing for this service. Scope and pricing depend on the amount of data to be migrated and the effort required. Data migration from NAS, AIX Virtual I/O (AIX VIO), or SAN Volume Controller environments requires special migration methods and must be identified and scoped.

### 3.2.2 IBM DS8000 Data Migration Services

DS8000 Data Migration Services migrate data from IBM ESS 800 to DS8000, or DS8000 to DS8000 storage systems. They use temporary licenses for copy services, allowing you to migrate projects without long-term license commitments.

**Value Proposition**
When you do not need long-term copy services, IBM DS8000 Data Migration Services allows you to use Copy Services functions for short-term migration scenarios. It allows enhanced migrations of data between disk subsystems without the long-term commitment and cost of licenses.

**Benefits**
Using IBM DS8000 Data Migration Services provides these benefits:

- Correct migration of data to the DS8000 Storage System
- Shorter migration schedule
- Expert storage architects
- Reduced risk of disruption to business

**Key Questions and Considerations**
Use following questions to determine whether IBM DS8000 Data Migration Services are right for you:

- Do you have the available skills to effectively migrate data to the IBM DS8000 Storage System?
- If you have the skills internally, would it be a better use of your resources to have IBM provide DS8000 migration services?
- Would you like to build DS8000 migration skills on your staff?
- Do you want to have your data migrated to your new IBM DS8000 Storage System by highly skilled and experienced specialists?

**Deal size and pricing**
Contact an STG Lab Services Opportunity Manager for an estimate of pricing for this service. Scope and pricing dependent on the amount of data to be migrated and the effort required.
3.2.3 Additional Services & Offerings

Additional Services and Offerings are available from STG Lab Services and details can be found by visiting the following web address:

http://www.ibm.com/systems/services/labservices/

Use the Contact Now link in the right corner to contact STG Lab Services. You will be directed to fill out a form.
Migration scenarios

This part provides detailed explanations and illustrations of various disk migration techniques and products.

This part contains the following chapters:

- Data migration using IBM Remote Mirror and Copy
- DSCLbroker
- IBM XIV data migration
- SAN Volume Controller-based migration
- Using mirroring techniques
- Using TDMF for z/OS
- z/OS Dataset Mobility Facility
- Using TDMF TCP/IP for z/OS
Data migration using IBM Remote Mirror and Copy

This chapter addresses the IBM System Storage DS8000 Remote Mirror and Copy methods used for data migration.

This chapter contains the following sections:

- IBM System Storage Remote Mirror and Copy overview
- DS8000 user interface for Copy Services
- Data migration from Enterprise Storage Server Model 800 to DS8000
- Data migration from DS8000 to DS8000
- Post-migration tasks
4.1 IBM System Storage Remote Mirror and Copy overview

The Copy Services functions of the DS8000 are a set of flexible data mirroring solutions that allow replication between volumes of disk storage systems. These functions are used to implement remote data backup and disaster recovery solutions.

The Remote Mirror and Copy functions are optional licensed functions of the DS8000 that include:

- Metro Mirror
- Global Copy
- Global Mirror
- Metro/Global Mirror

In addition, System z users can use the DS8000 for:

- z/OS Global Mirror
- z/OS Metro/Global Mirror

For more information about these topics, see the following books:

- *IBM System Storage DS8000: Copy Services in Open Environments*, SG24-6788
- *DS8000 Copy Services for IBM System z*, SG24-6787

This chapter specifically deals with two members of this family of products, Metro Mirror and Global Copy.

**Exception:** The Global Mirror member of the Remote Mirror and Copy family is not addressed because using IBM Global Mirror is not a practical approach for data migration. Global Mirror is intended to be a long-distance business continuity solution. The cost and setup complexity of IBM Global Mirror are not typically justified for a one-time data migration project.

Remote Mirror and Copy functions can be used to migrate data between IBM enterprise storage subsystems. These subsystems can include IBM Enterprise Storage Server® Model 800 and Model 750.

**Important:** IBM Enterprise Storage Server Model F20 is not supported for Remote Mirror and Copy functions to the DS8000. If you intend to migrate from the Model F20, you must move the data to a Model 800 or 750 before migrating to the DS8000.

Remote Mirror and Copy migration methods replicate data between IBM storage subsystems, and are sometimes called *hardware-based replication*. Remote Mirror and Copy migration offers the following advantages:

- High performance
- Operating system-independent
- Does not consume application host resources

Global Copy migration methods require you to stop application updates and changes while the remaining data is moved. Both Global Copy and Metro Mirror require the application host operating system to acquire the new storage subsystem volumes. The following factors must be taken into account when planning the data migration:

- Acquire new migration volumes and make them known to the operating system
- Modify application configuration files
- Perform data integrity checks
The steps for performing the data migration using Metro Mirror are similar to the steps required for Global Copy. For more information, see 4.3, “Data migration from Enterprise Storage Server Model 800 to DS8000” on page 51. Any differences required for each method are noted.

Appropriate license codes must be purchased and applied to each of the systems in the migration scenario to use the Remote Mirror and Copy functions. For information about licensing requirements, speak to your IBM storage marketing representative.

4.1.1 Metro Mirror

Metro Mirror, previously known as Synchronous Peer-to-Peer Remote Copy (PPRC), provides real-time mirroring of logical volumes between two supported storage subsystems (DS8000 and Enterprise Storage Server). These subsystems can up to 300 km apart from each other. It is a synchronous copy solution. Write operations must be completed on both local and remote copies before they are considered to be complete. It is typically used for applications that cannot suffer any data loss in the event of a failure. It can also be used effectively as a data migration tool. Figure 4-1 shows the sequence of a write update with Metro Mirror.

![Metro Mirror write sequence](image)

Figure 4-1 Metro Mirror write sequence

When the application performs a write update operation to a source volume, the following steps occur:

1. Write to source volume (storage unit cache and nonvolatile storage)
2. Write to target volume (storage unit cache and nonvolatile storage)
3. Signal write complete from the remote target DS8000
4. Post I/O complete to host server

The Fibre Channel connection between the local and the remote disk subsystems can be through a switch. Other supported distance solutions include dense wave division multiplexing (DWDM).
Metro Mirror performance considerations
Because Metro Mirror is a synchronous mirroring technology, it decreases performance for the write operations. However, unlike operating system mirroring, Metro Mirror does not consume any processor activity. Use bandwidth analysis and capacity planning to determine how many Metro Mirror links you need to ensure the best possible performance.

As part of your implementation project, identify and distribute hot spots across your configuration to manage and balance the load.

Keep in mind the following basic considerations:
- Is your Fibre Channel network bandwidth between storage subsystems too small, causing an increase in the response time of your applications at moments of high workload?
- Do not share the Metro Mirror link I/O ports with host attachment ports. Sharing ports can result in unpredictable performance of Metro Mirror, and make performance problem analysis more complicated.
- Distance is an important topic. Data writes must go to the secondary disk subsystem, and then be acknowledged. With latency times of active components on the way, plan for approximately a 1 ms delay per 100 km for write I/O.
- Sometimes the problem is not Metro Mirror, but rather hot spots on the disks. Be sure that these problems are resolved before you start with Metro Mirror. Invest the time necessary to plan the logical configuration of the target DS8000 to lay out the data. Be sure to plan for high I/O rates and future growth.

4.1.2 Global Copy
Global Copy, previously known as Peer-to-Peer Remote Copy Extended Distance (PPRC-XD), copies data non-synchronously for both open and z/OS systems. Global Copy is a great tool for data migrations, especially for longer distances than are supported with Metro Mirror.

When operating in Global Copy mode, the source volume sends a periodic, incremental copy of updated tracks to the target volume instead constant updates. Incremental updates cause less impact to application writes for source volumes and less demand for bandwidth resources, allowing a more flexible use of the available bandwidth.

With Global Copy, write operations complete on the source disk subsystem before they are received by the target disk subsystem. This capability is designed to prevent the local performance from being affected by wait time of writes on the remote system. Therefore, the source and target copies to be separated by any distance.
Figure 4-2 illustrates the write sequence for Global Copy.

The numbers in the figure refer to the following workflow:
1. The host server makes a write I/O to the source (local) storage unit. The write is staged through cache and nonvolatile storage.
2. The write returns as completed to the host server’s application.
3. The source storage unit sends the necessary data so that the updates are reflected on the target (remote) volumes. This is a non-synchronous process, so there is a delay. The updates are grouped in batches for efficient transmission.
4. The target returns write complete to the source storage unit when the updates are secured in the target DS8000 cache and nonvolatile storage. The source then resets its Global Copy change recording information.

Data Migration using Global Copy requires a consistent copy of the data to move applications and servers to the remote subsystem. By design, the data on the remote system is a fuzzy copy, with the volume pairs in a copy pending state. To create a consistent copy of the migrated data, the application must be quiesced and the volume pairs suspended. If the pairs are terminated before quiescing I/O and suspending the pairs, you might lose ordered transactions to the remote site.

There are two ways to ensure data consistency during migration using Global Copy:
- Shut down all applications at the primary site and allow the out-of-sync sectors drain completely.
- Issue the `go to sync` command on the Global Copy relationships when the out-of-sync sectors are approaching or at zero. When the out-of-sync sectors are fully drained, the pairs are in full duplex mode and there is a consistent relationship. Using `go to sync` is the more reliable method.
**Global Copy performance considerations**

Global Copy employs a two-phase process of transferring data from primary to auxiliary storage subsystems. The first phase transfers all existing data to an out of sync bitmap, and this data is transferred to the secondary. When this operation is complete, the volumes are flagged as first pass true. After the first pass, Global Copy looks at all the tracks that have changed since the last time it looked in a round robin fashion. Global Copy then moves this data into the out of sync bitmap, and transfers the new changes to the secondary disk subsystem.

Global Copy performance depends on the bandwidth of the Fibre Channel interface (the pipe) between the Global Copy primary and remote storage subsystems. During the first pass, Global Copy is of no practical use to data migration because the first pass must complete. If the data to be transferred is 5 TB and the WAN/DWDM interface operates at 27 MBps, allow about 60 hours for the first pass. This large amount of time is not a reflection on the performance of Global Copy. Instead, it reflects how long it takes to move 5 TB at 27 MBps.

The pipe must also be large enough to handle the write load of the migration volumes. Using the same pipe bandwidth previously mentioned, the amount of data being written to all the migration volumes should be less than 27 MBps. Otherwise, Global Copy will not be able to deliver the data to the remote storage quickly enough to keep up. As a result, the data at the remote site will fall further and further behind the data at the primary site. Again, this limitation is not a reflection on the capabilities of Global Copy. Make sure that the pipe is large enough to handle the write load of the migration candidate application.

Dedicate the host adapter ports used for Global Copy to the Global Copy migration. Host-based activities should *not* use the same ports as Global Copy because this would negatively affect Global Copy performance.

**4.1.3 Target DS8000 configuration considerations**

All the normal considerations for ensuring optimum performance of a new application need to be followed when considering the performance of a migrated application. For example, consider an existing Oracle application on a Enterprise Storage Server 800 that is coming off lease. In this example, the application is running well on 16 RAID-5 arrays composed of 15 K RPM 18 GB DDMs (1.68 TB). Once a week, the application is Flash Copied to another set of 16 identical RAID-5 arrays (another 1.68 TB). This application cannot be successfully deployed to a DS8000 with two 10 K RPM 300 GB RAID-5 Arrays (3.4 TB).

Although Global Copy would be able to successfully migrate the data, the application performance after the migration would surely be far less than on the DS8000. This is limited because the performance characteristics (I/O per second and MBps) of the 16 Enterprise Storage Server arrays far outweigh the performance capabilities of two DS8000 arrays.

The performance of a migrated application must be considered as though it were a new application. Use tools such as Disk Magic to properly size the storage. Also, use performance disciplines like good data layout for I/O balance across multiple arrays, device adapters, and host adapters. Also take into account considerations for FlashCopy performance.
4.1.4 Automation software: Tivoli Storage Productivity Center for Replication

The IBM Tivoli Storage Productivity Center for Replication is a useful tool for managing Copy Services products for the Enterprise Storage Server and DS8000. These products include Global Mirror, Global Copy, and FlashCopy. It also manages FlashCopy and Metro Mirror for the SAN Volume Controller. Tivoli Storage Productivity Center for Replication manages Copy Services in the following ways:

- Automating administration and configuration of these services with wizard-based sessions and copy set definitions
- Allowing control of copy services tasks, including starting, suspending, and resuming
- Offering tools for monitoring and managing copy sessions

If you are using Tivoli Storage Productivity Center for Replication to manage Copy Services, use it to facilitate data migration when using Copy Services as your migration technique.

If you are not using IBM Copy Services, do not use Tivoli Storage Productivity Center for Replication because it adds unnecessary cost and complexity to the implementation. For a one-time migration project using Copy Services, use Global Copy controlled by one of the DS8000 user interfaces.

4.2 DS8000 user interface for Copy Services

There are two native user interface methods for the DS8000: IBM System Storage DS8000 Storage Manager graphical user interface (DS GUI), and the DS Command Line Interface (DS8000 CLI). Both of these user interfaces can be used to start copy services functions. Additionally, System z Interfaces with TSO and ICKDSF can be used to start the DS8000 copy services functions.

4.2.1 IBM System Storage DS8000 Storage Manager

The IBM System Storage DS8000 Storage Manager (DS Storage Manager) software is installed on the Hardware Management Console (HMC) provided with the DS8000. The software can also be installed on a separate personal computer running Windows or Linux. The DS Storage Manager allows you to control Copy Services functions in real time. For specific information about installing on a personal computer, see the IBM System Storage DS8000 User's Guide, SC26-7915.

The DS Storage Manager can be accessed through the Tivoli Storage Productivity Center Element Manager of the System Storage Productivity Center (SSPC). It can be accessed from any network-connected workstation with a supported browser. It can also be accessed directly from the DS8000 management console using the browser on the HMC.

Remotely connect to the DS Storage Manager using the IP address or fully qualified name resolved by the DNS server. The correct port number is the address as shown in the following example.

https://10.0.0.1:8452/DS8000/Login

In this example, 10.0.0.1 is the IP address of the HMC. 8452 is the port number, and DS8000/Login is required to access the Storage Manager.
You are prompted for a user name and password (typically managed by the System Administrator) as shown in Figure 4-3.

![IBM System Storage DS8000 Sign On](image1)

Figure 4-3  DS8000 Storage Manager login window

The login window for newer versions of DS8000 is shown in Figure 4-4.

![IBM System Storage DS8000 Storage Manager](image2)

Figure 4-4  DS8000 new Storage Manager login window from release 6.1
For specific details about using the Storage Manager for data migration, see 4.3, “Data migration from Enterprise Storage Server Model 800 to DS8000” on page 51.

DS Storage Manager is currently supported by the following browsers:
- Internet Explorer 7.x and 8.x
- Mozilla Firefox 3.5, and 3.6

### 4.2.2 IBM System Storage DS8000 command-line interface

The DS8000 command-line interface (DS8000 CLI) provides a full set of commands to perform all copy services functions. The DS8000 CLI code is installed on one of the supported open systems platforms (AIX, Windows, Solaris, Linux, or HP-UX). Although DS8000 CLI is not installed on a System z server, it can create and manage System z volumes in a FlashCopy, and Remote Mirror and Copy environment. The DS8000 CLI can also create volumes for all supported open systems servers. Use the latest version of DS8000 CLI for data migration because it is compatible with earlier versions of the DS8000.

For information about how to create login profiles, see *IBM System Storage DS8000 Command Line Interface User’s Guide*, SC26-7916. Access can also be obtained by starting the DS8000 CLI using the TCP/IP address or the fully qualified name of the storage unit HMC. The user ID and password used for DS8000 CLI are the same as those used for the DS Storage Manager. For more details about using DS8000 CLI for Remote Mirror and Copy functions, see 4.4, “Data migration from DS8000 to DS8000” on page 81.

### 4.2.3 System z interfaces

Several System z interfaces can be used to manage the IBM System Storage DS8000 Copy Services functions in addition to the Storage Manager and DS8000 CLI. This section covers the following z/OS options:
- TSO
- ICKDSF

These interfaces send commands directly to the DS8000 storage unit over a FICON or ESCON channel to a conduit count key data (CKD) volume. The command is passed to the microcode for execution from this volume. The commands are issued the same way for DS8000 and Enterprise Storage Server.

#### TSO for Metro Mirror and Global Copy management

For z/OS, the TSO Metro Mirror and Global Copy commands offer a powerful and flexible interface to manage the Remote Mirror and Copy environment. TSO commands can also be used for control of open system volumes in a Metro Mirror environment. TSO commands require at least one CKD volume on the storage units to act as a conduit. TSO commands communicate with the DS8000 and Enterprise Storage Server through a device number specified with the command. IP connectivity is not required because the conduit is a FICON or ESCON channel. TSO commands can be integrated into REXX programs for automation purposes. Table 4-1 lists the available commands and descriptions.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CESTPAIR</td>
<td>Establishes Metro Mirror and Global Copy volume pairs</td>
</tr>
<tr>
<td>CESTPATH</td>
<td>Establishes Remote Mirror and Copy paths</td>
</tr>
</tbody>
</table>
Data Migration to IBM Disk Storage Systems

For a detailed description of these commands, see *z/OS DFSMS Advanced Copy Services*, SC35-0428, and *DS8000 Copy Services for IBM System z*, SG24-6787.

**ICKDSF for Metro Mirror and Global Copy management**

In System z environments, the ICKDSF utility offers a means of control for Metro Mirror and Global Copy functions. ICKDSF typically runs as a batch program, and can be automatically run from batch scheduling products. It also supports VM and VSE systems. All commands must be addressed to a device that is either online or offline to the system where the batch job is submitted.

The ICKDSF commands used for Metro Mirror and Global Copy management are shown in Table 4-2.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDELPAIR</td>
<td>Deletes volume pairs</td>
</tr>
<tr>
<td>CDELPATH</td>
<td>Deletes Remote Mirror and Copy paths</td>
</tr>
<tr>
<td>CGROUP</td>
<td>Controls volume groups with <strong>FREEZE</strong> and <strong>RUN</strong></td>
</tr>
<tr>
<td>CQUERY</td>
<td>Queries the status of the volumes and paths</td>
</tr>
<tr>
<td>CRECOVER</td>
<td>Recovers data on the recovery system</td>
</tr>
<tr>
<td>CSUSPEND</td>
<td>Suspends Metro Mirror and Global Copy volume pairs</td>
</tr>
<tr>
<td>PPRCOPY ESTPATH</td>
<td>Establishes Metro Mirror and Global Copy paths</td>
</tr>
<tr>
<td></td>
<td>between a primary and secondary LSS</td>
</tr>
<tr>
<td>PPRCOPY ESTPAIR</td>
<td>Establishes Metro Mirror and Global Copy</td>
</tr>
<tr>
<td></td>
<td>volume pairs</td>
</tr>
<tr>
<td>PPRCOPY DELPATH</td>
<td>Deletes Metro Mirror and Global Copy paths</td>
</tr>
<tr>
<td></td>
<td>between a primary and secondary LSS</td>
</tr>
<tr>
<td>PPRCOPY RECOVER</td>
<td>Allows a system to regain control of a volume on</td>
</tr>
<tr>
<td></td>
<td>the secondary DS8000</td>
</tr>
<tr>
<td>PPRCOPY SUSPEND</td>
<td>Puts a Metro Mirror or Global Copy pair in the</td>
</tr>
<tr>
<td></td>
<td>suspended state</td>
</tr>
<tr>
<td>PPRCOPY FREEZE</td>
<td>Suspends all Metro Mirror or Global Copy operations</td>
</tr>
<tr>
<td></td>
<td>at the LSS level</td>
</tr>
<tr>
<td>PPRCOPY RUN</td>
<td>Resume I/O operations after a freeze with Extended</td>
</tr>
<tr>
<td></td>
<td>Long Busy</td>
</tr>
<tr>
<td>PPRCOPY QUERY</td>
<td>Queries the Metro Mirror or Global Copy volume</td>
</tr>
<tr>
<td></td>
<td>pair and path status</td>
</tr>
</tbody>
</table>

For more information, see the *Device Support Facilities User’s Guide and Reference*, GC35-0033.
4.3 Data migration from Enterprise Storage Server Model 800 to DS8000

Data migration from Enterprise Storage Server to DS8000 using Metro Mirror or Global Copy requires the following codes:

- PPRC V2 feature code on the Enterprise Storage Server
- Remote Mirror and Copy license codes on the DS8000

This section highlights the steps required to set up the configuration to migrate data from Enterprise Storage Server to DS8000. It includes paths and pairs using DS Storage Manager, DS8000 CLI, TSO, and ICKDSF for Metro Mirror, and Global Copy. It also includes the specific steps to complete the data migration from an open systems or a System z perspective. The final actions to attach the host to the target DS8000 and start the applications are covered in 4.5, “Post-migration tasks” on page 90.

The following steps are covered:

- Adding Enterprise Storage Server Copy Services Domain to DS8000
- Creating Remote Mirror and Copy paths
- Creating Remote Mirror and Copy pairs
- Completing the data migration

To establish PPRC pairs, the Enterprise Storage Server must have a PPRC license and the DS8000 must have a Remote Mirror Copy license. The Enterprise Storage Server must also have Fibre Channel adapters that have connectivity to the DS8000.

**Important:** To manage the Enterprise Storage Server Copy Services from the DS8000, you must have your IBM Service Representative install Licensed Internal Code Version 2.4.3.65 or later on the Enterprise Storage Server. You also need DS8000 code bundle 6.0.500.52 or later on the DS8000.

4.3.1 Adding Enterprise Storage Server Copy Services Domain to the DS8000 Storage Complex

To migrate data using the DS Storage Manager from the Enterprise Storage Server to the DS8000, add the Enterprise Storage Server Copy Services Domain server to the DS8000 Storage Complex. Adding the server can only be done using the DS8000 Storage Manager. The DS8000 CLI does not require authentication like the DS Storage Manager does.

The DS8000 Storage Manager must authenticate to the Enterprise Storage Server before you can issue any commands. The user ID and password used to log on to the DS Storage Manager must be defined in the Enterprise Storage Server Specialist.
In Figure 4-5, the DS Storage Manager user ID is SLT-TEAM, which requires that the user ID and password be created using the Enterprise Storage Server Specialist.

To create the user ID and password, perform the following steps:

1. Click Enterprise Storage Server Specialist from the Enterprise Storage Server Launch window. The Enterprise Storage Server Specialist window opens as shown in Figure 4-5.

2. Click Users. The User Administration window opens (Figure 4-6).

3. Click Modify Users to add, remove, or modify existing user accounts.
4. Add the user ID and password that matches the DS Storage Manager user ID and password as seen in Figure 4-7. Click **Perform Configuration** to create the user.

![Figure 4-7 Enterprise Storage Server Specialist Modify Users panel](image1.png)

The new user is displayed in the User Administration window as shown in Figure 4-8.

![Figure 4-8 Enterprise Storage Server Specialist User Administration panel with DS Storage Manager ID](image2.png)

5. Connect to the Storage Manager and click **Real-time manager → Manage Hardware → Storage complexes**.
6. In the Storage Complex window, click **Select Action → Add 2105 Copy Services Domain**. A window prompting for the 2105 CS Domain Server 1 IP Address (Server A) is displayed (Figure 4-9).

![IBM System Storage DS8000 Storage Manager](image)

**Figure 4-9** Adding a 2105 Copy Services Domain to the DS8000 Storage Complex

7. Enter the Server 1 address. You can enter a Server 2 IP addresses by checking the **Define a second Copy Services Server** check box. Click **OK**.

8. Verify that the CS Domain has been added to the Storage Complex by viewing the Storage Complexes after the task completes (Figure 4-10).

![IBM System Storage DS8000 Storage Manager](image)

**Figure 4-10** 2105 Copy Services Domain added to DS8000 Storage Complex

The remaining steps for setting up Metro Mirror paths can be done with both the Storage Manager and the DS8000 CLI. Both methods are shown in 4.3.2, “Creating Remote Mirror and Copy paths” on page 54.

### 4.3.2 Creating Remote Mirror and Copy paths

Creating paths for Metro Mirror or Global Copy for data migration requires physical connections between the Enterprise Storage Server and the DS8000. These paths can be through a switch (including hardware for extended distances), or through direct physical connections between the storage units.
In Metro Mirror, the Fibre Channel links can be connected by up to two switches. If the paths are created through a switch, the ports for the storage units must be in an active zone set. The zone configuration can be completed in several ways. In this example, two ports from each storage unit are put in a single zone (Brocade) and activated as seen in Figure 4-11.

![Figure 4-11 Enterprise Storage Server and DS8000 zone members](image)

**Remember:** If channel extension technology is used for Metro Mirror links, make sure the product used is supported in the environment (direct connect or SAN switch). Also ensure that the SAN switch used is supported by the product vendor.

You need to know what the physical-to-logical layout of the I/O ports on both storage units is so that you can set up the paths correctly. The chart in Figure 4-12 shows the numbering scheme for the ports on the Enterprise Storage Server. There are four host bays, each with four slots for adapters. The Enterprise Storage Server can have up to 16 host adapters, allowing for a maximum of 16 Fibre Channel ports per Enterprise Storage Server.

![Figure 4-12 Enterprise Storage Server I/O enclosures](image)

The chart in Figure 4-13 on page 56 displays the scheme for the DS8000 in the first frame. There are four I/O enclosures, each with six slots for adapters. Two slots in each enclosure are reserved for the device adapters connected to the disk drive enclosures. That leaves four slots for host adapters.
These adapters can either be ESCON or Fibre Channel. The Fibre Channel adapters can be either 2 Gbps or 4 Gbps, and have four ports each. The slot numbers as shown Figure 4-13 on page 56 are logical slot numbers as seen in either DS Storage Manager or DS8000 CLI.

The logical number is one less than the physical number. The four ports on the Fibre Channel adapters are labeled 0-3, and the numbering starts at the top port on each adapter.

Figure 4-13  DS8000 first expansion frame I/O enclosures
The DS8000 second expansion frame I/O enclosures are shown in Figure 4-14.

Creating Remote Mirror and Copy paths with DS Storage Manager

After the physical paths are set up, the Metro Mirror/Global Copy paths can be created. To create paths using DS Storage Manager, perform the following steps:

1. From the DS Storage Manager, click **Real Time Manager → Copy Services → Paths**.

2. In the Paths: Real-time window shown in Figure 4-15, select the Enterprise Storage Server you added in 4.3.1, “Adding Enterprise Storage Server Copy Services Domain to the DS8000 Storage Complex” on page 51 from the **Storage complex** list.

3. Select the Source LSS from the **Storage Unit** list.
4. Click the **Select Action** box and select **Create**. The paths are being set up from the Enterprise Storage Server to the DS8000 for the data migration. In this example, four paths exist for LSS 17 (this area is empty if paths do not exist).

**Tip:** Selecting the source LSS is optional because the next window requires the LSS selection. If it is selected, any existing paths are displayed.

5. In the Select source LSS window shown in Figure 4-16, select the source LSS you want and click **Next**.

**Remember:** Each LSS to be used in the data migration requires paths to be created one at a time using DS Storage Manager or the DS8000 CLI. A source LSS can have multiple target LSSs, but the paths must be created for each target LSS separately.

In this example, LSS 17 is selected as the source LSS. Only the LSSs on the Enterprise Storage Server are displayed in this list.

![Create paths: Real-time](image-url)

*Figure 4-16  Create Paths - Source LSS*
6. In the Select target LSS window, select the target LSS and click **Next**. Select the target Storage Complex, Storage unit, Storage image, and LSS from the lists. Then select the storage unit to generate the list of compatible LSSs to be displayed for selection.

   In this example, LSS 10 is selected for this storage unit (Figure 4-17).

   ![Select Target LSS for the DS8000](image)

   **Figure 4-17** Select Target LSS for the DS8000

7. In the Select source I/O ports window as shown in Figure 4-18, select one or more source ports and click **Next**.

   The ports listed are available in the active zone set in the switch, or on the paths physically connected between the two storage units. Figure 4-18 on page 59 shows that the ports listed for the Enterprise Storage Server (Source) are the same as the zone members shown in Figure 4-11 on page 55.

   The ports are displayed by location rather than WWPN.

   ![Select PPRC ports on Enterprise Storage Server](image)

   **Figure 4-18** Select PPRC ports on Enterprise Storage Server

8. In the Select target I/O ports window as shown in Figure 4-19, select at least one target port for each source port and click **Next**. As with the source ports, the target ports available depends on how the zoning or cabling is set up. To select multiple target ports for a single source port, press the Shift key while selecting the ports.
Each Source I/O port has a path available to both Target I/O ports due to the way the zone was established on the switch. Both ports from both storage units are in a single zone. With this selection, there are four logical paths for LSS 17 on the Enterprise Storage Server, to LSS 10 on the DS8000.

Tip: Have more than one path available for bandwidth and redundancy reasons. The workload is balanced across the available paths by the code. Isolating the paths from host I/O paths guarantees no interference from host I/O.

For Metro Mirror, use multiple paths due to timing sensitivity issues. Global Copy does not have this sensitivity to shared host I/O ports and paths.

Each Source I/O port has a path available to both Target I/O ports due to the way the zone was established on the switch. Both ports from both storage units are in a single zone. With this selection, there are four logical paths for LSS 17 on the Enterprise Storage Server, to LSS 10 on the DS8000.

Tip: Have more than one path available for bandwidth and redundancy reasons. The workload is balanced across the available paths by the code. Isolating the paths from host I/O paths guarantees no interference from host I/O.

For Metro Mirror, use multiple paths due to timing sensitivity issues. Global Copy does not have this sensitivity to shared host I/O ports and paths.

9. In the Select path options window (Figure 4-20 on page 61), select Define as consistency group and click Next.

For Remote Mirror and Copy pairs, selecting the consistency group option supports the consistent data between two LSSs (not a group of LSSs). Data consistency means that the sequence of dependent writes is always kept in the copied data.

Tip: The consistency group option is not required for Global Copy paths in a data migration scenario.

The Define as consistency group option itself can keep consistent data at the remote site. In a rolling disaster, all volumes go into the queue full condition within the time interval specified in the Consistency Group time-out value. The default time-out value is 120 seconds.

However, if all the volumes do not go into the queue full condition, use the commands freezepprc and unfreezepprc to hold the I/O activity to the volumes not in the queue full condition. You can also resume or release the held I/O without waiting for the Consistency Group timeout to minimize the impact on the applications. These commands are issued at the LSS level through the DS8000 CLI.
For more information about using consistency groups, see *IBM System Storage DS8000: Copy Services in Open Environments*, SG24-6788, and *DS8000 Copy Services for IBM System z*, SG24-6787.

**Remember:** The `freezepprc` command removes the Remote Mirror and Copy paths between the source LSS and target LSS.

10. In the Verification window shown in Figure 4-21, review your selections carefully and then click **Finish**. If changes need to be made, click **Back** to make the modifications before clicking **Finish**.

![Figure 4-20  Consistency groups defined during PPRC paths creation](image)

![Figure 4-21  Confirm the selections](image)
11. After the selections are correct and the command completes, the created paths are displayed as shown in Figure 4-22. Paths can be removed or added if necessary.

![Figure 4-22](image.png)

**Creating Remote Mirror and Copy paths with DS8000 CLI**

The Remote Mirror and Copy paths can also be created by connecting to the Enterprise Storage Server using the DS8000 CLI. Log on to the Enterprise Storage Server using an existing user ID and password on the Enterprise Storage Server, or using the one created for DS Storage Manager.

Start the DS8000 CLI using the IP address of one cluster on the Enterprise Storage Server and enter the user ID and password as shown in Example 4-1. A list of the storage units in the Copy Services Domain Server is displayed. In this example, there are two storage units.

**Example 4-1  Connect to Enterprise Storage Server using DS8000 CLI**

```
C:\Program Files\ibm\dscli>dscli -hmc1 9.155.51.201
Enter your username: SLE-TEAM
Enter your password:
Date/Time: July 12, 2011 11:34:04 AM CET IBM DSCLI Version: 5.2.410.182  DS:
IBM.2105-22665 IBM.2105-22673
```

To create Remote Mirror and Copy paths, perform the following steps:

1. Before the paths can be created, you need to determine the remote WWNN and the available paths. Use the `lssi` command to determine the remote WWNN as shown in Example 4-2.

**Example 4-2  Using the lssi command to obtain the WWNN of the remote system**

```
dscli> lssi
Date/Time: July 12, 2011 11:46:06 AM CET IBM DSCLI Version: 5.2.410.182
Name       ID               Storage Unit     Model WWNN             State  ESSNet
==================================================================================
DS8k-SLE05 IBM.2107-75L4741 IBM.2107-75L4740 931 5005076305FFC786 Online Enabled
```
2. Display the available PPRC ports between the two storage units using the `lavailpprcport` command as seen in Example 4-3. The `-remotewwnn` parameter is the WWNN determined in the previous step. Use the Enterprise Storage Server (2105 in this example) as the `-dev` parameter, and the DS8000 (2107 in this example) as the `-remotedev` parameter. This information matches what was seen for the DS Storage Manager. They are identical because of the active zone created at the beginning of this section and shown in Figure 4-11 on page 55.

Example 4-3 Query available PPRC ports between Enterprise Storage Server and DS8000

dscli> lsavailpprcport -dev IBM.2105-22673 -remotedev IBM.2107-75L4741
-remotewwnn 5005076305FFC786 -fullid 17:10
Date/Time: July 6, 2011 10:02:47 AM CET IBM DSCLI Version: 5.2.410.182 DS:
IBM.2105-22673
Local Port           Attached Port          Type
================================================
IBM.2105-22673/I0004 IBM.2107-75L4741/I0140 FCP
IBM.2105-22673/I0004 IBM.2107-75L4741/I0142 FCP
IBM.2105-22673/I00AC IBM.2107-75L4741/I0140 FCP
IBM.2105-22673/I00AC IBM.2107-75L4741/I0142 FCP

3. Use the WWNN and a port pair to create a path using the `mkpprcpath` command from the Enterprise Storage Server (Example 4-4). Verify that the information entered is correct before running the command. In this example, four paths are created between LSS 17 on the Enterprise Storage Server and LSS 10 on the DS8000.

Example 4-4 Creating PPRC paths between Enterprise Storage Server and DS8000

dscli> mkpprcpath  -dev IBM.2105-22673 -remotedev IBM.2107-75L4741 -remotewwnn
5005076305FFC786 -srclss 17 -tgtlss 10 I0004:I0140 I00AC:I0140 I0004:I0142 I00AC:I0142
Date/Time: July 6, 2011 10:11:39 AM CET IBM DSCLI Version: 5.2.410.182 DS:
IBM.2105-22673
CMUC00149I mkpprcpath: Remote Mirror and Copy path 17:10 successfully
established.

4. Query and verify the paths using the `lspprcpath` command as shown in Example 4-5. If you need to make changes, remove the path and recreate or modify it using DS Storage Manager. The output lists the Enterprise Storage Server information in the `Src`, `SS`, and `Port` columns, and the DS8000 information in the `Tgt` column.

Example 4-5 Query PPRC paths between Enterprise Storage Server and DS8000

dscli> lspprcpath -dev IBM.2105-22673 17
Date/Time: July 6, 2011 10:19:04 AM CET IBM DSCLI Version: 5.2.410.182 DS:
IBM.2105-22673
Src Tgt State   SS   Port  Attached Port Tgt WWNN
=========================================================  17 10 Success FF10 I00AC I0140 5005076305FFC786
17 10 Success FF10 I00AC I0142 5005076305FFC786
17 10 Success FF10 I0004 I0140 5005076305FFC786
17 10 Success FF10 I0004 I0142 5005076305FFC786
Creating paths with TSO
To establish a path between the Enterprise Storage Server and the DS8000 for Metro Mirror, use the CESTPATH command. You must know the SSID, WWNN, and LSS number for the primary and remote storage units. These numbers can be displayed with the CQUERY command.

The example shown in Example 4-6 establishes a path between the Enterprise Storage Server and the DS8000. The SSID of the primary volume is x'1710', the WWNN is 5005076300C09629, and the LSS is x'00'. The SSID of the secondary volume is x'1711', the WWNN is 5005076305FFC786, and the LSS is x'01'. In this example, the consistency group option is set to NO.

Example 4-6  TSO CESTPATH output

<table>
<thead>
<tr>
<th>CESTPATH DEVN(X'2028') -</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRIM(X'1710' 5005076300C09629 X'00') -</td>
</tr>
<tr>
<td>SEC(X'1711' 5005076305FFC786 X'01') -</td>
</tr>
<tr>
<td>LINK(X'00300200') CGROUP(NO)</td>
</tr>
</tbody>
</table>

Creating paths with ICKDSF
The PPRCOPY ESTPATH command is used to establish Remote Mirror and Copy paths between the primary and remote LSSs. Each command can establish up to eight paths.

In Example 4-7, the FCPP parameter specifies up to eight paths. Each path is an 8-digit hexadecimal address in the form x'aaaaabbbGa67 b'. In this form, aaaa is the primary system adapter (SAID) and bbbb is the remote system adapter ID (SAID). The World Wide Node Name (WWNN) for the primary and remote are specified in the WWNN parameter, with the primary listed first followed by the remote.

Example 4-7  ICKDSF ESTPATH output

```bash
//IKJEFT01 JOB MSGLEVEL=(1,1),MSGCLASS=A,NOTIFY=user id
//STEP01 EXEC PGM=ICKDSF
//SYSPRINT DD SYSOUT=*  
//VOL1 DD UNIT=3390,VOL=SER=DS6400,DISP=SHR  
//SYSIN DD *
PPRCOPY ESTPATH DDNAME(VOL1) -
  FCPP(X'00000100') PRI(X'0002',AAVCA) -
  SEC(X'0003',AAVCA) WWNN(005076300C09629,5005076305FFC786) -
  CGROUP(NO) LSS(X'00',X'01')
```

4.3.3 Creating Remote Mirror and Copy pairs
When you are creating pairs for Metro Mirror or Global Copy, you must take into account the size of the volumes to be copied from the Enterprise Storage Server. The DS8000 default volume is created as 2^30 bytes, but the Enterprise Storage Server volumes are created as 10^9 bytes.

When creating fixed block volumes on the DS8000, you have three size choices:
- ds: The number of bytes allocated will be the requested capacity value times 2^{30}.
- Enterprise Storage Server: The number of bytes allocated will be the requested capacity value times 10^9.
- blocks: The number of bytes allocated will be the requested capacity value times 512 bytes (each block is 512 bytes).
To create the pairs, the volume on the DS8000 must be as large or larger than the volume on the Enterprise Storage Server. If the volumes on the DS8000 are larger than the Enterprise Storage Server, the extra space is not used. However, if you might use the Enterprise Storage Server as a remote system after the migration, make the volumes the same size. For fixed block volumes, create the DS8000 volumes using the Enterprise Storage Server type.

For more information, see *IBM System Storage DS8000: Copy Services in Open Environments*, SG24-6788.

CKD volumes have the same considerations about volume size. CKD volumes are specified in number of cylinders. The volumes on the DS8000 must have the same or greater number of cylinders as the volumes on the Enterprise Storage Server.

For more information, see *DS8000 Copy Services for IBM System z*, SG24-6787.

Another important aspect to consider before creating the configuration for the data migration to the DS8000 is the volume address differences between the two storage units. On the Enterprise Storage Server, open systems volume IDs are given in an 8-digit format, xxx-sssss. In this form, xxx is the LUN ID and sssss is the serial number of the Enterprise Storage Server. When referring to these volumes with DS8000 CLI, add 1000 to the volume ID. Remember the limitations on the Enterprise Storage Server address ranges shown in Figure 4-23.

![Figure 4-23](image)

On the DS8000, the range of available addressing is significantly greater than on the Enterprise Storage Server. The entire storage unit can be configured for just CKD or just FB using LUNs 00-FF on LSSs/LCUs 00-FE (FF is reserved).

If the configuration on the DS8000 is a mixed CKD and FB environment, the CKD or FB volumes must be contained within a grouping of 16 LSSs/LCUs. For example, CKD volumes are configured in 2000-2FFF and FB volumes in 3000-3FFF, where 20-2F and 30-3F are the range of 16 LCUs/LSSs.

No restrictions exist for the creation of the LSS/LCU ranges other than to group by 16. The 16 groups can be all CKD, all FB, or mixed, as long as any single group is the same type. Take grouping into account when planning the DS8000 configuration because the storage unit enforces the groupings.

After you complete the volume configuration on the DS8000 compatible with the Enterprise Storage Server, the volumes will be formatted by an internal DS8000 process. Otherwise the volumes cannot be used as target volumes on the DS8000. This formatting must be complete before creating the pairs. The time needed for the volume initialization completion varies depending on the size.

**Remember:** If you attempt to use the volumes *before* the volume initialization has completed, the establish of the copy pairs fails. This is an expected result in this case.
Creating Remote Mirror and Copy pairs with DS Storage Manager

To create the Remote Mirror and Copy pairs using the DS Storage Manager, perform the following steps:

1. Open the Metro Mirror wizard using PROCEDURE.

2. In the initial window of the wizard (Figure 4-24), select the Storage Complex, the Storage Unit, and the Resource Type (LSS or Show All Volumes).

3. If you selected LSS in the Resource Type list, select the specific LSS in the Specify LSS list. If you selected Show All Volumes, select All FB Volumes or All CKD Volumes.

4. Select the volumes to be used, and click the Select Action list and select Create.

   In this example, the selection is being made by LSS. Specify LSS 17 (which we have already made paths for) volume 1700.

5. In the Volume Pairing Method window, select the method for Volume Pairing:

   - Automated volume pair assignment automatically pairs the first selected source volume with a target volume of the same size. All subsequent pairs are automatically assigned based on compatible size in a sequential fashion. The lowest source volume number are paired with the lowest target volume number.
Manual volume pair assignment requires a manual selection of all source and target volumes. This method is used in the example (Figure 4-25). Repeat the process for each selected source volume.

6. The Select source volumes window displays the available source volumes based on an LSS. You can also create paths from this panel by clicking Create Paths, which starts the Paths wizard.

In this example, the source volumes are in LSS 17 on the Enterprise Storage Server. Select the source volume you want and click Next (Figure 4-26).
7. In the Source ID window, select the target Storage complex, Storage unit, Storage image, and Resource type as shown in Figure 4-27. The example uses the DS8000 LSS 10 volume 1010 as the target device.

8. Select the volume for the pair and click OK.

9. In the Select copy options window, select the option for creating the pairs as Metro Mirror or Global Copy (Figure 4-28).
Additional options are available for selection in this panel:

- Reset reservation (Open Systems)
- Perform initial copy
- Permit read access from target

The remaining three options have a dependency on the type of copy options selected. **Perform initial copy** allows the selection of Suspend Metro Mirror relationships after initial copy. The remaining two options are available only for CKD pairs.

Click **Next** after making your selections.

10. The Verification window prompts for confirmation of the selected configuration of pairs. If you need to make any changes, click **Back** and make them before returning to this window. When ready, click **Finish** to complete the pair selection as seen in Figure 4-29.

![Figure 4-29 Confirm the selection](image)

11. After the pairs have been created, verify the state of the relationship by checking the **Metro Mirror: Real-time** window for the defined source volumes (Figure 4-30).

![Figure 4-30 Check state of the pairs](image)

Remote Mirror and Copy pairs are created one LSS at a time. All the volumes in an LSS can be used, but the process must be repeated for each LSS involved in the migration.
Creating Remote Mirror and Copy pairs with DS8000 CLI

After the Remote Mirror and Copy paths are created, create Metro Mirror and Global Copy pairs between the Enterprise Storage Server and the DS8000. Use the mkpprc command to create pairs as shown in Example 4-8. In this example, a single Metro Mirror pair is created between source volume 1700 on the Enterprise Storage Server and target volume 1010 on the DS8000.

Example 4-8  mkpprc with DS8000 CLI

dscli> mkpprc -dev IBM.2105-22673 -remotedev IBM.2107-75L4741 -type mmir 1700:1010
Date/Time: July 7, 2011 10:34:18 AM CET IBM DSCLI Version: 5.2.410.182 DS: IBM.2105-22673
CMUC00153I mkpprc: Remote Mirror and Copy volume pair relationship 1700:1010 successfully created.

Requirement: The DS8000 CLI must be used from the Enterprise Storage Server to create the Remote Mirror and Copy pairs.

Example 4-9 illustrates the creation of Global Copy pairs with DS8000 CLI on the same volumes used in the preceding examples. Before starting this example, the Metro Mirror pair was removed.

Example 4-9  Create Global Copy pairs

dscli> mkpprc -dev IBM.2105-22673 -remotedev IBM.2107-75L4741 -type gcp 1700:1010
Date/Time: July 13, 2011 11:19:22 AM CET IBM DSCLI Version: 5.2.410.182 DS: IBM.2105-22673
CMUC00153I mkpprc: Remote Mirror and Copy volume pair relationship 1700:1010 successfully created.

Creating the Remote Mirror and Copy pairs can be tedious task if many volumes are involved in the migration. Up to 4096 source volumes are possible on the Enterprise Storage Server. An easy way to create many pairs is to use the DS8000 CLI in a scripting mode. A script to create Global Copy pairs is shown in Figure 4-31.

```bash
#> cat mkpprc_script
# PPRC for IBM1->IBM2 CKD pairs
dscli -cfg IBM1_IBM2 -user admin -passwd xxxxx mkpprc -dev IBM.2105-22673
-remotedev IBM.2107-75L4741 -type gcp 0000-003f:0000-3f 0100-013f:0100-3f 0200-023f:0200-3f
0300-033f:0300-3f 0400-043f:0400-3f 0500-053f:0500-3f 0600-063f:0600-3f 0700-073f:0700-3f
0800-083f:0800-3f 0900-093f:0900-3f 0a00-0a3f:0a00-3f 0b00-0b3f:0b00-3f
0c00-0c3f:0c00-3f 0d00-0d3f:0d00-3f 0e00-0e3f:0e00-3f 0f00-0f3f:0f00-3f
#>
```

Figure 4-31  DS8000 CLI script example for mkpprc

This example shows creating 1024 Global Copy pairs. The source volumes are on the Enterprise Storage Server (which has 64 volumes in each LCU) and pairs are created with corresponding target volumes on the DS8000. The pairs are created as a single pair of LSSs at a time. For example, source LSS 00 with volumes 00-3F is paired with target LSS C8 with volumes 00-3F as 0000-003F:C800-C83F.
Creating pairs with TSO

The CESTPAIR command is used to establish Metro Mirror and Global Copy relationships between the Enterprise Storage Server and the DS8000. This command is used to indicate what kind of an operation to perform:

- An initial establish of volumes that were in the simplex state
- A resynchronization of a suspended pair of volumes
- A Failover/Failback process

In this example, you are establishing an initial copy from a simplex state.

The option parameter OPTION has two mutually exclusive values: SYNC and XD. SYNC is specified to create Metro Mirror pairs. XD is used to specify Global Copy pairs.

In Example 4-10, the primary volume is in LSS x'00' on the Enterprise Storage Server. The primary volume has the following characteristics:

- The SSID of the LSS is x'1710'
- The serial number of the Enterprise Storage Server is 22673
- The CCA is x'28'
- The LSS is x'00'

The remote volume is in LSS x'01' on the DS8000, and has the following characteristics:

- The SSID of the LSS is x'1711'
- The serial number of the DS8000 is L4741
- The CCA is x'28'
- The LSS is x'01'.

The MSGREQ(YES) parameter specifies that Metro Mirror waits until the initial full volume copy operation is complete before issuing the completion message.

Example 4-10   TSO CESTPAIR Metro Mirror

| CESTPAIR DEVN(X'2028') - PRIM(X'1710' 22673 X'28' X'00') - SEC(X'1711' L4741 X'28' X'01') - OPTION(SYNC) MODE(COPY) ONLINSEC(NO) MSGREQ(YES) |

In order for a pair to be created as Global Copy, specify whether the pair comes from the simplex or suspended state. This means an initial copy of a newly established pair (simplex) or a resynchronization of a suspended pair. The MODE parameter is used to specify either COPY or RESYNC. The CESTPAIR command in Example 4-11 includes the OPTION(XD) and MODE(COPY) to signify Global Copy from a simplex state.

Example 4-11   TSO CESTPAIR Global Copy

| CESTPAIR DEVN(X'400A') OPTION(XD) MODE(COPY) - PRIM(X'4000' ABTV1 X'0A' X'00') - SEC(X'8000' 20781 X'8A' X'80') |
Creating pairs with ICKDSF

The ESTPAIR command is used to establish Metro Mirror and Global Copy relationships between the Enterprise Storage Server and the DS8000 volumes. In Example 4-12, the pair is created for Metro Mirror (the OPTION(SYNC) parameter). To create the pair as Global Copy, use OPTION(XD).

Example 4-12 ISCKDSF ESTPAIR

```plaintext
//IKJEFT01 JOB MSGLEVEL=(1,1),MSGCLASS=A,NOTIFY=user id
//STEP01 EXEC PGM=ICKDSF
//SYSPRINT DD SYSOUT=* 
//VOL1 DD UNIT=3390,VOL=SER=DS8100,DISP=SHR
//SYSIN DD*
PPRCOPY ESTPAIR DDNAME(VOL1) PRI(X'0002',AAVCA,X'00')- SEC(X'0003',22673,X'00') MODE(COPY) MSGREQ(NO) OPTION(SYNC) - CRIT(NO) LSS(X'00',X'01')
```

### 4.3.4 Completing the data migration

This section includes the steps taken with DS8000 CLI, TSO, ICKDSF, and DS Storage Manager to move the production site on to the new storage. Moving the host systems is covered in 4.5, “Post-migration tasks” on page 90.

After the Remote Mirror and Copy pairs are established, the data will start copying. For Metro Mirror, wait for the pairs to enter a Full Duplex state so that the data migration can be completed. For Global Copy, there is an intermediate state required to get to the Full Duplex state.

The following steps synchronize the data from the Enterprise Storage Server to the DS8000. These steps are common for any interface used to manage the Remote Mirror and Copy relationships.

1. Verify full duplex mode (Metro Mirror) or out-of-sync tracks (Global Copy) are at or near zero.
2. If you are using Global Copy, convert to synchronous (go-to-sync function).
3. Suspend the source I/O.
4. Suspend pairs.
5. Delete pairs.

**Completing the data migration with the DS8000 CLI**

To complete the migration with the DS8000 CLI, perform the following steps:

1. Use the DS8000 CLI command `lspprc` to query the state of the Metro Mirror pairs as shown in Example 4-13. Only one pair is shown in this example. The current state is Copy Pending, which means the copy is in progress. The copy is complete when the state changes to Full Duplex.

Example 4-13 Monitor state of Metro Mirror pairs

```plaintext
dsc1i> lspprc -dev IBM.2105-22673 1700
Date/Time: July 8, 2011 10:35:49 AM CET IBM DSCLI Version: 5.2.410.182 DS: IBM.2105-22673
ID State Reason Type SourceLSS Timeout (secs) Critical Mode First Pass Status
==================================================================================
```

72 Data Migration to IBM Disk Storage Systems
The same command is used to query the Global Copy relationship. Query the pairs to monitor the **Out of Sync Tracks**, as seen in Example 4-14. In this example, the local and remote copies still have many tracks to copy (288991) and the **First Pass** has not yet completed.

**Example 4-14  Non-zero Out of Sync Tracks**

dscli> lspprc -1 -dev IBM.2105-22673 1700
Date/Time: July 14, 2011 4:07:53 PM CET IBM DSCLI Version: 5
ID        State        Reason Type Out Of Sync Tracks First Pass Status
===============================================================================
1700:1010 Copy Pending - Global Copy 288991 .......... False

As this number approaches zero (or gets to zero) and the **First Pass** field becomes true, the **go to sync** function is started. This state must be reached for each pair in the data migration. Example 4-15 shows the **Out of Sync Tracks** at zero (0) and the **First Pass Status** at true.

**Example 4-15  Out of Sync Tracks at zero**

dscli> lspprc -1 -dev IBM.2105-22673 1700
Date/Time: July 14, 2011 4:17:40 PM CET IBM DSCLI Version:
ID        State        Reason Type Out Of Sync Tracks First Pass Status
===============================================================================
1700:1010 Copy Pending - Global Copy 0 .......... True

3. This Global Copy pair is now ready for the final copy to be performed. Before running this command, stop the host I/O and synchronize all the data to disk to avoid losing any updates after the relationship is removed.

4. Use the **mkpprc** command to start the **go to sync** function on the same pair, but change the **-type** option to `mmir` (Example 4-16).

**Example 4-16  Go to Sync function**

dscli> mkpprc -dev IBM.2105-22673 -remotedev IBM.2107-75L4741 -type mmir 1700:1010
Date/Time: July 14, 2011 4:18:16 PM CET IBM DSCLI Version: 5.2.410.182 DS: IBM.2105-22673
CMUC00153I mkpprc: Remote Mirror and Copy volume pair relationship 1700:1010 successfully created.

5. Query the pair using the **lspprc** command to verify that the pair has completed the copy when the state changes to **Full Duplex** (Example 4-17).

**Example 4-17  Full Duplex state**

dscli> lspprc -1 -dev IBM.2105-22673 1700
Date/Time: July 14, 2011 4:18:35 PM CET IBM DSCLI Version:
ID        State        Reason Type Out Of Sync Tracks
===============================================================================
1700:1010 Full Duplex - Metro Mirror 0
6. The copy is now complete and the I/O is all stopped. Remove the relationship with the `rmpprc` command as shown in Example 4-18.

**Tip:** Using this command with multiple pairs in multiple LSSs will result in a confirmation question for each range of pairs. The `-quiet` option can be included to turn off this confirmation prompt.

**Example 4-18  Remove the relationship**

```
dscli> rmpprc -dev IBM.2105-22673 -remotedev IBM.2107-75L4741 1700:1010
Date/Time: July 15, 2011 10:35:58 AM CET IBM DSCLI Version: 5.2.410.182 DS:
IBM.2105-22673
CMUC00160W rmpprc: Are you sure you want to delete the Remote Mirror and Copy
volume pair relationship 1700:1010:? [y/n]:y
CMUC00155I rmpprc: Remote Mirror and Copy volume pair 1700:1010 relationship
successfully withdrawn.
```

7. You are asked to confirm the removal of the pair. Respond `y` (for yes) to delete the relationship, or `n` (for no) to cancel.

Complete these steps for all pairs involved in the data migration.

**Completing the migration with TSO**

To complete the migration with TSO, perform the following steps:

1. The CQUERY command is used to query the status of one volume of a pair. It can also be used to query all paths that are associated with the LSS for the specified device number. This command can be issued to either a primary or remote volume. A host system attached to only the primary volume cannot obtain the status of the secondary volume for that pair.

The output listed in Example 4-19 is from the CQUERY command issued to volume `x'400A'`. The command was issued specifically for a Global Copy pair.

Notice that the **STATE** is **PENDING.XD** to signify Global Copy. In addition, the Out of Sync Tracks are listed in this example as **119475**. The total number of tracks on the volume and the Percent Copy Complete are listed below.

This information is **not** listed for volumes with all tracks in sync. The path information is shown only when the CQUERY command is issued to a primary volume. In this example, two paths are viable.

Although the example is for a Global Copy relationship, a Metro Mirror pair displays equivalent information during the copy phase. However, the **STATE** is displayed as **PENDING** instead of **PENDING.XD**.

**Example 4-19  TSO CQUERY command output**

```
ANTP8802I CQUERY DEVN(X'400A')
ANTP0090I CQUERY FORMATTED LVL 3 881
VOLUME REPORT
************** PPRC REMOTE COPY CQUERY - VOLUME *******************
* (PRIMARY) (SECONDARY) *
* SSID CCA LSS SSID CCA LSS *
*DEVICE LEVEL STATE PATH STATUS SERIAL# SERIAL# *
*-------- ------- --------- --------- --------- *
* 400A PRIMARY... PENDING.XD ACTIVE.. 4000 0A 00 8000 8A 80 *
* CRIT(NO)....... CGRPLB(NO). 00000000BT1 0000000020781*
```

Tip: Using this command with multiple pairs in multiple LSSs will result in a confirmation question for each range of pairs. The `-quiet` option can be included to turn off this confirmation prompt.
Chapter 4. Data migration using IBM Remote Mirror and Copy

2. Query the status of the target volumes using the CQUERY command as shown in Example 4-20. Notice that the state of the pair is PENDING.XD and no path information is displayed. This is a normal response for the target storage unit.

Example 4-20   Query target status

2. Before removing relationships, stop all host activity and ensure that all data is written to disk. These steps ensure that a full and complete copy exists on the DS8000. After that, move the pairs to a synchronous state, complete the copy, and attach the application to the target storage unit.
4. The pair is ready for synchronization after all the tracks have been copied. Use the
CESTPAIR command with the SYNC option on the same pair (Example 4-21).

Example 4-21  CESTPAIR to synchronize the pair

CESTPAIR DEVN(X'400A') OPTION(SYNC) MODE(RESYNC) -
PRIM(X'4000' ABTV1 X'0A' X'00') -
SEC(X'8000' 20781 X'8A' X'80')

Monitor the copy, using the CQUERY command until the pair reaches a Duplex state. When
this state is reached, the copy is complete.

5. Use the CSUSPEND command to suspend and remove the pairs as shown in Example 4-22.
Query the state to confirm that the pairs are suspended using the CQUERY command.
The state will be SUSPEND(3), which means the Global Copy was suspended by a host
cmd to the source storage unit.

Example 4-22  CQUERY to suspend the pair

CSUSPEND DEVN(X'400A') -
PRIM(X'4000' ABTV1 X'0A' X'00') -
SEC(X'8000' 20781 X'8A' X'80')

6. The copy is now complete and the pairs are suspended. Delete the pairs by issuing the
CDELPPAIR command to the source storage volumes (Example 4-23).

Example 4-23  CDELPPAIR to delete the pair

CDELPPAIR DEVN(X'400A') -
PRIM(X'4000' ABTV1 X'0A' X'00') -
SEC(X'8000' 20781 X'8A' X'80')

After these steps are performed for all pairs involved in the data migration, you are ready to
move the host and application to the target DS8000.

Completing the migration with ICKDSF
To complete the migration with ICKDSF, perform the following steps:

1. Use the PPRCOPY QUERY command to check the state of pairs as shown in
Example 4-24.

Example 4-24  ICKDSF query pair

//IKEF01 JOB MSGLEVEL=(1,1),MSGCLASS=A,NOTIFY=user id
//STEP01 EXEC PGM=ICKDSF,TIME=30 ,PARM='NOREPLYU'
//SYSIN DD DDNAME(Vol1)
//SYSIN DD DDNAME(Vol1) PATHS

When the pair reaches either of the following states, you are ready to suspend the host
application and flush all writes to disk:

If the pair created is a Metro Mirror relationship and reaches DUPLEX state, or it is a Global
Copy relationship and the out-of-sync tracks number is near zero,

2. A Global Copy relationship must be converted to synchronous to complete the copy. Use
the PPRCOPY CESTPAIR command, changing the parameter OPTION(XD) to
OPTION(SYNC) and adding the parameter MODE(RESYNC). The relationship is shown in Example 4-25.

**Example 4-25  ICKDSF go-to-sync**

```plaintext
PPRCOPY ESTPAIR UNIT(4080) LSS(X'01',X'03') -
PRI(X'A001',22673,X'00') SEC(X'2801',L4741,X'00') -
OPTION(SYNC) MODE(RESYNC)
```

3. Use the `PPRCOPY SUSPEND` command to suspend the Metro Mirror relationship before you remove it as shown in Example 4-26. Removing the mirror ensures all the data has been copied from the source to the target storage unit.

**Example 4-26  ICKDSF suspend pair**

```plaintext
//IKJEFT01 JOB MSGLEVEL=(1,1),MSGCLASS=A,NOTIFY=userid
//STEP01 EXEC PGM=ICKDSF
//SYSPRINT DD SYSOUT=* 
//VOL1 DD UNIT=3390, VOL=SER=DS8100, DISP=SHR 
//SYSIN DD * 
PPRCOPY SUSPEND DDNAME(VOL1) PRI(X'0002',22673,X'00') -
   SEC(X'0003',AAVCA,X'00') LSS(X'00',X'01')
```

4. Delete the pair using `PPRCOPY DELPAIR` command as shown in Example 4-27.

**Example 4-27  ICKDSF delete pair**

```plaintext
//IKJEFT01 JOB MSGLEVEL=(1,1),MSGCLASS=A,NOTIFY=userid
//STEP01 EXEC PGM=ICKDSF
//SYSPRINT DD SYSOUT=* 
//VOL1 DD UNIT=3390, VOL=SER=DS8100, DISP=SHR 
//SYSIN DD * 
PPRCOPY DELPAIR DDNAME(VOL1) PRI(X'0002',22673,X'00') -
   SEC(X'0003',AAVCA,X'00') LSS(X'00',X'01')
```

After performing these steps for all pairs involved in the data migration, you are ready to move the host and application to the target DS8000.

**Completing the migration with DS Storage Manager**

To complete the data migration using DS Storage Manager, complete the following steps:

1. To monitor the out-of-sync tracks, click **Real-time manager → Copy services → Metro Mirror → Select storage specifics**.
2. Select the pair to monitor and click the **Out-of-sync tracks** option.
Figure 4-32 shows an example of querying a pair for out-of-sync tracks. The out-of-sync tracks are at zero (0). This means that you are ready to complete the data migration from the storage units and move the applications to the new storage.

![Figure 4-32](image)

**Figure 4-32** Zero out-of-sync tracks

The example in Figure 4-33 shows the out-of-sync tracks as non-zero. Monitor this state until it gets to the state shown in Figure 4-32.

![Figure 4-33](image)

**Figure 4-33** Non-zero out-of-sync tracks
3. For a Global Copy relationship, select the pairs that are now ready to convert to sync (out-of-sync tracks are at zero or close to zero). Using the same path previously mentioned, select the **Convert to synchronous** option in the **Select Action** list as shown in Figure 4-34.

![Figure 4-34   Convert to synchronous](image)

4. In the next window, select which system to suspend the Global Copy relationship at and select **Suspend** from the **Select Action** list (Figure 4-35).

![Figure 4-35   Suspend Metro Mirror](image)
5. In the Select source or target for volumes to be suspended window, select **Suspend at source** as shown in Figure 4-36.

![Suspend Metro Mirror: Real-time](image)

**Figure 4-36** Suspend at source selected

6. Click **OK** to confirm the selection as shown in Figure 4-37.

![Convert to synchronous: Real-time](image)

**Figure 4-37** Confirm the selection

7. After the command completes, query the state of the relationship. As seen in Figure 4-38, the pairs are now in full duplex state. After stopping the host I/O and confirming that all the out-of-sync tracks are at zero, you are ready to remove the relationship.

![Metro Mirror: Real-time](image)

**Figure 4-38** Metro Mirror Full Duplex

**Important:** Do not remove the relationship if the out-of-sync tracks are non-zero. Removing the pairs any earlier would result in having incomplete data on the remote system.
8. Select the pair involved in the migration, and select **Suspend** from the **Select Action** list.

9. Select the pair involved in the migration again, and select **Delete** the **Select Action** list as shown in Figure 4-39.

![Figure 4-39 Delete Metro Mirror pairs](image)

A confirmation warning for the deletion displays (Figure 4-40).

![Figure 4-40 Confirm Delete](image)

The data migration using IBM Copy Services is complete. The remaining steps for bringing the applications up on the new DS8000 are listed in 4.5, “Post-migration tasks” on page 90.

### 4.4 Data migration from DS8000 to DS8000

The steps needed for migrating data between two DS8000s are similar to those for migrating data from Enterprise Storage Server to DS8000. The Copy Services steps can be started from either the local or a remote storage unit.

If the DS Storage Manager is used for data migration, the storage unit of one system must be added to the storage complex of the other. This process is described in 4.3.1, “Adding Enterprise Storage Server Copy Services Domain to the DS8000 Storage Complex” on page 51.
The steps needed for moving the application to the new storage are the same regardless of what the source storage unit is. For more information, see 4.5, “Post-migration tasks” on page 90.

Because the commands for Enterprise Storage Server and DS8000 as source storage are similar, the migration between two DS8000s is covered using the DS8000 CLI only. The following steps are described:

- Configuring the remote DS8000
- Creating Remote Mirror and Copy paths
- Creating Remote Mirror and Copy pairs
- Completing the data migration from DS8000 to DS8000

### 4.4.1 Configuring the remote DS8000

The remote storage unit can be configured by using DS Storage Manager (Real-time and Simulated), or by using DS8000 CLI. The volumes on the remote system must match or be larger than the volume sizes and types on the local system. As noted previously, if the target volumes are created larger than the source volumes, the extra space is not used. In addition, if the system that the data is being migrated from will ever be used as a Remote Mirror and Copy remote for the new DS8000, the volumes must have matching sizes.

The examples in this section use DS8000 CLI for performing the configuration. To configure the remote DS8000, follow these steps:

1. Confirm the logical configuration on the source storage unit using the `lsfbvol` command as shown in Example 4-28. For simplicity reasons, the fb volumes in volume group V3 are listed as the source volumes. Notice that the capacities of these volumes are 5 GB and 1 GB (DS sizes).

   **Example 4-28  Check the logical configuration on local**
   
   ```
   dscli> lsfbvol -volgrp v3 -dev IBM.2107-75ABTV1 4204-4206
   Date/Time: July 8, 2011 1:16:40 PM PDT IBM DSCLI Version: 7.6.10.511 DS: IBM.2107-75ABTV1
   Name       ID     accstate  datastate configstate deviceMTM  datatype extpool cap (2^30B)
   TEAMRED_A  4204 Online  Normal  Normal      2107-900  FB  512   P4    5.0
   TEAMRED_A  4205 Online  Normal  Normal      2107-900  FB  512   P4    1.0
   TEAMRED_A  4206 Online  Normal  Normal      2107-900  FB  512   P4    1.0
   ```

2. Issue the `mkfbvol` command to configure the target DS8000 to match the source DS800 as shown in Example 4-29. You need to issue the command once for each size of volume.

   In this example, the first command creates two 5 GB volumes with the same nickname (pprc_tgt_tic6) and assigns both to the volume group V20. The second command is the same except for the size (1 GB).

   **Example 4-29  Create matching configuration on remote**
   
   ```
   dscli> mkfbvol -dev IBM.2107-7520781 -extpool p0 -cap 5 -type ds -name pprc_tgt_tic6 -volgrp v20 E404-E405
   Date/Time: July 8, 2011 1:30:45 PM PDT IBM DSCLI Version: 7.6.10.511 DS: IBM.2107-7520781
   CMUC000251 mkfbvol: FB volume E404 successfully created.
   CMUC000251 mkfbvol: FB volume E405 successfully created.
   ```
dscli> mkfbvol -dev IBM.2107-7520781 -extpool p0 -cap 1 -type ds -name pprc_tgt_tic6 -volgrp v20 E406

Date/Time: July 8, 2011 1:31:25 PM PDT IBM DSCLI Version: 7.6.10.511 DS: IBM.2107-7520781
CMUC00025I mkfbvol: FB volume E406 successfully created.

Reminder: Volumes created with DS8000 CLI in a single command will be the same size and from the same extent pool. Volumes with even LSSs are created in even extent pools. Volumes with odd LSSs are created in odd extent pools.

3. Verify the configuration on the target DS8000 using the `lsfbvol` command (Example 4-30).

Example 4-30  Verify the logical configuration on the remote

dscli> lsfbvol -dev IBM.2107-7520781 E404-E406

Date/Time: July 8, 2011 1:33:49 PM PDT IBM DSCLI Version: 7.6.10.511 DS: IBM.2107-7520781
Name   ID   accstate datastate configstate deviceMTM datatype extpool cap (2^30B)
=================================================================================
pprc_tgt_tic6 E404 Online   Normal    Normal      2107-900  FB 512   P0  5.0
pprc_tgt_tic6 E405 Online   Normal    Normal      2107-900  FB 512   P0  5.0
pprc_tgt_tic6 E406 Online   Normal    Normal      2107-900  FB 512   P0  1.0

Repeat these steps until the configuration of the target DS8000 matches the configuration on the volumes used in the data migration. Remember to plan the layout of the volumes based on performance considerations and opportunities for growth.

4.4.2 Creating Remote Mirror and Copy paths

To create the Remote Mirror and Copy paths for the data migration using the DS8000 CLI, perform the following steps:

1. Use the `lssi` command to discover the device ID and WWNN of the remote system as shown in Example 4-31.

   The information listed under ID is the device ID that will be used. In this example, the device ID is IBM.2107-7520781. The model is 922. The WWNN is 5005076303FFC1A5.

Example 4-31  Query device ID and WWNN using lssi command

dscli> lssi

Date/Time: July 8, 2011 2:09:22 PM PDT IBM DSCLI Version: 7.6.10.511 DS: -
Name   ID   Storage Unit   Model   WWNN             State  ESSNet
=================================================================================
-  IBM.2107-7520781 IBM.2107-7520780 922   5005076303FFC1A5 Online Enabled

If the remote system is a logically partitioned (LPAR) storage unit, the same `lssi` command can be used. However, it returns the information for both LPARs (also known as Storage Facility Images or SFI). You must know which SFI the data will be migrated to.
The output for an LPAR system is shown in Example 4-32. Here the device IDs are IBM.2107-75ABTV1 and IBM.2107-75ABTV2. The model is 9A2, with both WWNNs listed for each SFI.

Example 4-32  Query device ID and WWNN on an LPAR storage unit

dscli> lssi
Date/Time: July 8, 2011 2:12:34 PM PDT IBM DSCLI Version: 7.6.10.511 DS: -
Name ID                   Storage Unit     Model WWNN             State  ESSNet
=================================================================================
-  IBM.2107-75ABTV1 IBM.2107-75ABTV0 9A2   5005076303FFC663 Online Enabled
-  IBM.2107-75ABTV2 IBM.2107-75ABTV0 9A2   5005076303FFCE63 Online Enabled

2. After the zoning and connectivity are ready, query which paths are available to use for Remote Mirror and Copy using the lsavailpprcport command. Issue this command from the local system to the remote.

In Example 4-33, the command queries the possible ports between the local system IBM.2107-75ABTV1 and the remote system IBM.2107-7520781, between LSS 42 and E4. As you can see in the output, there are two available paths configured. Notice that each attached port (remote) is visible to each local port.

Example 4-33  Check for available paths between the storage units

dscli> lsavailpprcport -dev IBM.2107-75ABTV1 -remotedev IBM.2107-7520781
-remotewwnn 5005076303FFC1A5 42:E4
Date/Time: July 8, 2011 2:26:04 PM PDT IBM DSCLI Version: 7.6.10.511 DS: IBM.2107-75ABTV1
Local Port Attached Port Type
============================
10011      I0400         FCP
10012      I0402         FCP

3. Create the Remote Mirror and Copy paths between the two storage units using the DS8000 CLI command mkpprcpath. In Example 4-34, paths are created between two different ports on each storage unit, 10011:10400 and 10012:10402 ( ). These paths are created using the same physical adapter on the local system. For more information, see “Global Copy performance considerations” on page 46.

Example 4-34  Create pprc paths with mkpprcpath command

dscli> mkpprcpath -dev IBM.2107-75ABTV1 -remotedev IBM.2107-7520781 -remotewwnn 5005076303FFC1A5 -srclss 42 -tgtlss E4 10011:10400 10012:10402
Date/Time: July 8, 2011 3:04:56 PM PDT IBM DSCLI Version: 7.6.10.511 DS: IBM.2107-75ABTV1
CMUC00149I mkpprcpath: Remote Mirror and Copy path 42:E4 successfully established.

Important: Remote Mirror and Copy paths need to be created between each LSS on the local and remote storage units for use in the data migration.
4. Verify that the paths were created correctly using the `lspprcpath` command as shown in Example 4-35 to confirm the path configurations. This command is issued against the local LSS 01. The output from this command lists the two paths you created in step 2.

**Example 4-35  Query the Remote Mirror and Copy paths with lspprcpath**

```
dscli> lspprcpath -dev IBM.2107-75ABTV1 42
Date/Time: July 8, 2011 3:11:36 PM PDT IBM DSCLI Version: 7.6.10.511 DS:
IBM.2107-75ABTV1
Src Tgt State   SS   Port  Attached Port Tgt WWNN
==========================================================================
42  E4  Success FFE4 I0011 I0400         5005076303FFC1A5
42  E4  Success FFE4 I0012 I0402         5005076303FFC1A5
```

### 4.4.3 Creating Remote Mirror and Copy pairs

Creating the Remote Mirror and Copy relationships between DS8000 storage units is the same as between Enterprise Storage Server and DS8000. For more information, see 4.3.3, “Creating Remote Mirror and Copy pairs” on page 64. Example 4-36 shows creating 64 global copy pairs in interactive mode. The success messages from each pair are included for first four relationships. The pairs can be created as Metro Mirror by substituting `mmir` for `gcp` when `-type` is specified.

**Example 4-36  Create Remote Mirror and Copy pairs using mkpprc**

```
dscli> mkpprc -dev IBM.2107-75ABTV1 -remotedev IBM.2107-7520781 -type gcp -mode full 4204-4243:E404-E443
Date/Time: July 8, 2011 3:27:51 PM PDT IBM DSCLI Version: 7.6.10.511 DS:
IBM.2107-75ABTV1
CMUC00153I mkpprc: Remote Mirror and Copy volume pair relationship 4204:E404 successfully created.
CMUC00153I mkpprc: Remote Mirror and Copy volume pair relationship 4205:E405 successfully created.
CMUC00153I mkpprc: Remote Mirror and Copy volume pair relationship 4206:E406 successfully created.
CMUC00153I mkpprc: Remote Mirror and Copy volume pair relationship 4207:E407 successfully created.
```

### 4.4.4 Completing the data migration from DS8000 to DS8000

Moving data from one DS8000 to another DS8000 is the same as moving data from Enterprise Storage Server to DS8000. For more information, see 4.3.4, “Completing the data migration” on page 72. The only difference is that the source and target DS8000 are displayed with all the information in the DS Storage Manager windows.
Figure 4-41 shows an example with three storage images available in the **Select storage complex** list: ATS_20780, DS8k_TIC06v1_ATS and DS8k_TIC06v2_SLE. The last two storage images belong to the same storage complex, which is logically partitioned (LPAR).

Figure 4-41   **DS storage complex with two DS8000s**

Figure 4-42 shows the new DS Storage Manager GUI.

Figure 4-42   **Mirroring Connectivity view from release 6.1**

Perform the following steps to complete the data migration using IBM Copy Services:

1. Monitor the copy progress using the **lspprc -l** command. The **-l** option is used to view the out-of-sync tracks. The command and output are displayed in Example 4-37.

   This output is formatted to highlight important information, specifically the out-of-sync tracks and the First Pass Status.

   **Example 4-37   Query the status of the Remote Mirror and Copy pairs using lspprc -l**
   
   ```
   dscli> lspprc -dev IBM.2107-75ABTV1 4208-4243
   Date/Time: July 11, 2011 3:25:18 PM PDT IBM DSCLI Version: 7.6.10.511 DS: IBM.2107-75ABTV1
   ID        State        Reason Type        Out Of Sync Tracks ... First Pass Status
   ===================================================================================
   4208:E408 Copy Pending -      Global Copy 100892 ... ... ... False
   ```

86   **Data Migration to IBM Disk Storage Systems**
Chapter 4. Data migration using IBM Remote Mirror and Copy

Query the state of the relationships until all pairs have zero (or near-zero) out-of-sync tracks and the First Pass Status is True (Example 4-38).

Example 4-38  Monitor for wanted state

dscli> lspprc -dev IBM.2107-75ABTV1 4204-4207
Date/Time: July 11, 2011 3:46:40 PM PDT IBM DSCLI Version: 7.6.10.511 DS: IBM.2107-75ABTV1
ID        State        Reason Type        Out Of Sync Tracks ... First Pass Status
==================================================================================
4204:E404 Copy Pending - Global Copy 0                  ... True
4205:E405 Copy Pending - Global Copy 0                  ... True
4206:E406 Copy Pending - Global Copy 0                  ... True
4207:E407 Copy Pending - Global Copy 0                  ... True

2. After this state is reached, the application is stopped and all data is written to disk. Convert the Global Copy to a synchronous copy by issuing the mkpprc command using mmir as the type as shown in Example 4-39.

Example 4-39  Go-to-sync with mkpprc

dscli> mkpprc -dev IBM.2107-75ABTV1 -remotedev IBM.2107-7520781 -type mmir 4204-4207:E404-E407
Date/Time: July 11, 2011 3:53:43 PM PDT IBM DSCLI Version: 7.6.10.511 DS: IBM.2107-75ABTV1
CMUC00153I mkpprc: Remote Mirror and Copy volume pair relationship 4204:E404 successfully created.
CMUC00153I mkpprc: Remote Mirror and Copy volume pair relationship 4205:E405 successfully created.
CMUC00153I mkpprc: Remote Mirror and Copy volume pair relationship 4206:E406 successfully created.
CMUC00153I mkpprc: Remote Mirror and Copy volume pair relationship 4207:E407 successfully created.

3. Monitor the state of the relationships using the lspprc -1 command until all pairs reach a full duplex state. Notice in Example 4-40 that First Pass Status is changed to Invalid, which is the expected output for a Metro Mirror relationship. This example shows both Copy Pending and Full Duplex states.

Example 4-40  Pairs in Copy Pending and Full Duplex states

dscli> lspprc -dev IBM.2107-75ABTV1 -remotedev IBM.2107-7520781 -1 4204-4207:E404-E407
Date/Time: July 11, 2011 4:28:08 PM PDT IBM DSCLI Version: 7.6.10.511 DS: IBM.2107-75ABTV1
ID        State        Type         Out Of Sync Tracks First Pass Status
==========================================================================
4204:E404 Copy Pending Metro Mirror 0                    Invalid
4205:E405 Copy Pending Metro Mirror 0                    Invalid
4206:E406 Copy Pending Metro Mirror 0                    Invalid
4207:E407 Full Duplex Metro Mirror 0                    Invalid
Continue to monitor the state until all relationships reach the Full Duplex state as shown in Example 4-41.

**Example 4-41  Check the state of the pairs with lspprc**

dscl> lspprc -dev IBM.2107-75ABTV1 4204-4207
Date/Time: July 11, 2011 4:31:20 PM PDT IBM DSCLI Version: 7.6.10.511 DS: IBM.2107-75ABTV1
===============================================================================
4204:E404 Full Duplex - Metro Mirror 0 ... ... ... ... .... Invalid
4205:E405 Full Duplex - Metro Mirror 0 ... ... ... ... .... Invalid
4206:E406 Full Duplex - Metro Mirror 0 ... ... ... ... .... Invalid
4207:E407 Full Duplex - Metro Mirror 0 ... ... ... ... .... Invalid

4. After the copy is complete, remove the Remote Mirror and Copy relationships using rmpprc command. Example 4-42 shows removing the 64 relationships. You can use the -quiet option as shown to disable the confirmation prompt.

**Example 4-42  Remove Remote Mirror and Copy relationships**

dscl> rmpprc -quiet -dev IBM.2107-75ABTV1 -remotedev IBM.2107-7520781 4204-4243:E404-E443
Date/Time: July 11, 2011 4:40:08 PM PDT IBM DSCLI Version: 7.6.10.511 DS: IBM.2107-75ABTV1
CMUC00155I rmpprc: Remote Mirror and Copy volume pair 4204:E404 relationship successfully withdrawn.
CMUC00155I rmpprc: Remote Mirror and Copy volume pair 4205:E405 relationship successfully withdrawn.
CMUC00155I rmpprc: Remote Mirror and Copy volume pair 4207:E407 relationship successfully withdrawn.

5. Query the relationships to confirm the removal by issuing the lspprc command. If the relationships are all removed, the command indicates that no Remote Mirror and Copy relationships were found as shown in Example 4-43.

**Example 4-43  Confirm Remote Mirror and Copy relationships are removed**

dscl> lspprc -dev IBM.2107-75ABTV1 -l 4204-4243
Date/Time: July 11, 2011 4:42:19 PM PDT IBM DSCLI Version: 7.6.10.511 DS: IBM.2107-75ABTV1
CMUC00234I lspprc: No Remote Mirror and Copy found.

If many LSSs are being used in the data migration, you might want to create a DS8000 CLI script to create paths and pairs. Automation allows you to verify all the information you are passing to DS8000 CLI in a single place, and avoid any typing mistakes. A script can also be used to set up queries to the pairs. These queries will let you know when to synchronize Global Copy, or remove the pairs when full duplex state is reached.
Figure 4-43 provides a sample script that starts the go-to-sync function for Global Copy. This example runs the command on 16 LSSs with 64 volumes each. The remote LSSs are spread across two ranges: C8-CF and E8-EF. The script can easily be modified for new scripts to query the pprc and remove the pairs.

```
#script to establish execute the go-to-sync function
mkpprc -dev IBM.2107-75ABTV1 -remotedev IBM.2107-7520781 -type mmir 0000-003f:c800-c83f
mkpprc -dev IBM.2107-75ABTV1 -remotedev IBM.2107-7520781 -type mmir 0100-013f:c900-c93f
mkpprc -dev IBM.2107-75ABTV1 -remotedev IBM.2107-7520781 -type mmir 0200-023f:ca00-ca3f
mkpprc -dev IBM.2107-75ABTV1 -remotedev IBM.2107-7520781 -type mmir 0300-033f:cb00-cb3f
mkpprc -dev IBM.2107-75ABTV1 -remotedev IBM.2107-7520781 -type mmir 0400-043f:cc00-cc3f
mkpprc -dev IBM.2107-75ABTV1 -remotedev IBM.2107-7520781 -type mmir 0500-053f:cd00-cd3f
mkpprc -dev IBM.2107-75ABTV1 -remotedev IBM.2107-7520781 -type mmir 0600-063f:ce00-ce3f
mkpprc -dev IBM.2107-75ABTV1 -remotedev IBM.2107-7520781 -type mmir 0700-073f:cf00-cf3f
mkpprc -dev IBM.2107-75ABTV1 -remotedev IBM.2107-7520781 -type mmir 0800-083f:ce00-ee3f
mkpprc -dev IBM.2107-75ABTV1 -remotedev IBM.2107-7520781 -type mmir 0900-093f:ee00-e93f
mkpprc -dev IBM.2107-75ABTV1 -remotedev IBM.2107-7520781 -type mmir 0a00-0a3f:e900-ea3f
mkpprc -dev IBM.2107-75ABTV1 -remotedev IBM.2107-7520781 -type mmir 0b00-0b3f:ea00-eb3f
mkpprc -dev IBM.2107-75ABTV1 -remotedev IBM.2107-7520781 -type mmir 0c00-0c3f:eb00-ec3f
mkpprc -dev IBM.2107-75ABTV1 -remotedev IBM.2107-7520781 -type mmir 0d00-0d3f:ec00-ed3f
mkpprc -dev IBM.2107-75ABTV1 -remotedev IBM.2107-7520781 -type mmir 0e00-0e3f:ed00-ee3f
mkpprc -dev IBM.2107-75ABTV1 -remotedev IBM.2107-7520781 -type mmir 0f00-0f3f:ef00-ef3f
```

Start the script, using the `-hmc` option with the name or IP address of the HMC on the primary system as shown in Example 4-44. You can also have a profile file set up. Place the name of the script after `-script` option followed by the `-user` and `-passwd` options. In this case, it is placed from a Windows XP Professional system.

**Example 4-44  Running a DS8000 CLI script**

```
C:\Program Files\ibm\dscli>dscli -hmc 9.155.62.102 -script gotosync.txt -user data -passwd whateveritis
```

The data migration using IBM Copy Services is now complete.
4.5 Post-migration tasks

The final steps for accessing the data after the IBM Copy Services data migration is complete vary by operating system. See the appropriate section for more details:

- AIX post-migration tasks
- Solaris and VxVM post-migration tasks
- System z post-migration tasks

4.5.1 AIX post-migration tasks

This subject is documented in Appendix A of IBM System Storage DS8000 Series: Copy Services in Open Environments, SG24-6788, which is available at:


The `recreatevg` command is covered in the Information Center:


This section addresses considerations and clarification details not covered in those references.

There are two phases to accessing an AIX volume group (VG) after an IBM Copy Services copy operation:

1. Configuring the LUNs. Depending on the multi-path code, the LUNs can be hdisks, vpaths, or something else if non-IBM storage is used.
   This step is accomplished by assigning the LUNs to a host from the storage side, then running `cfgmgr` at the host.

2. Accessing the VG.
   Import the VG with `importvg`. You can also run `recreatevg` against the LUNs if another copy (which can be the original VG) of the VG is located on the host.

   Use `recreatevg` if a copy of the VG is already on the host because the disks will have duplicate physical volume IDs (PVIDs), logical volume names, and file system mount points. The AIX Logical Volume Manager (LVM) does not allow duplicates of these characteristics. AIX does allow duplicate PVIDs on a system, but only as a temporary situation.

If there is a LUN varied on in a VG with PVID, you cannot configure another hdisk/vpath on the system with the same PVID using `cfgmgr`. To configure the disks in such a situation, you must vary off the VG with the duplicate PVID, and then run `cfgmgr`.

You can also configure the LUN on AIX before the copy of data is placed on the LUN (which creates the duplicate PVID). After the LUN is configured on AIX, subsequent FlashCopies or Global Copies will not require the LUN to be configured again. The LUN will not have a PVID because it has just been created.

Tip: It is not necessary to issue the following commands for clearing and setting a new PVID on the LUNs:

```
# chdev -l <hdisk#> -a pv=clear
# chdev -l <hdisk#> -a pv=yes
```

These commands are run by the `recreatevg` command, so they can be skipped.
In the following example, you have an existing VG \texttt{existingvg} defined on the system, and want to create a FlashCopy of it called \texttt{flashcopyvg}. To create the FlashCopy, perform the following steps:

1. Create target LUNs on the storage and assign them to the host.
2. Configure the LUNs on the host with \# \texttt{cfgmgr} (assuming you are using SDDPCM on a DS8000, which results in one hdisk for each LUN).
3. Make a note of the new hdisks (in the example they are hdisk10, hdisk11, and hdisk12).
4. Initiate the FlashCopy. If the file systems are not unmounted, see “Maintaining file system and data consistency” on page 91.
5. Run \texttt{recreatevg} to clean up the duplicate PVIDs, LV names, and so on, as follows:
   \# recreatevg -y flashcopyvg hdisk10 hdisk11 hdisk12
6. Vary on the VG with \# \texttt{varyonvg flashcopyvg}.

   \textbf{Note:} In this example, the new hdisks were configured before creating the FlashCopy, so they do not have a PVID. However, if you perform the FlashCopy first, you need to vary off \texttt{existingvg} before step 2.

If you want to update the copies later to match the VG again, follow these steps:

1. Vary off flashcopyvg using \# \texttt{varyoffvg flashcopyvg}.
2. Export flashcopyvg with \# \texttt{exportvg flashcopyvg}. This command removes information about flashcopyvg from the ODM on AIX, but does not change any information on the disks in the VG.
3. Create the copy with FlashCopy. For more information about when the application is not quiesced and the file systems are not unmounted, see “Maintaining file system and data consistency” on page 91.
4. Run \texttt{recreatevg} to create a VG from the FlashCopy volumes with new LV names using the format \# \texttt{recreatevg -y flashcopyvg hdisk10 hdisk11 hdisk12}. In this example, it is called \texttt{flashcopyvg}. This command also changes the PVIDs of the disks to unique PVIDs and loads the ODM with this information.

   \textbf{Tip:} You do not need to run \texttt{importvg} because the \texttt{recreatevg} command loads the ODM with the VG information.

For more information, see the recreatevg man page at:

5. Vary on the VG using \# \texttt{varyonvg flashcopyvg}.

   \textbf{Remember:} In this example, step 2 is necessary because you are going to rerun \texttt{recreatevg}, and you cannot have the VG already defined in the ODM.

You can make multiple copies of the VG using these steps.

\textbf{Maintaining file system and data consistency}
To maintain file system and data consistency, unmount the file systems because otherwise, data might be in file system cache and not be written to disk.
If the application is running, make sure the FlashCopy (or whatever disk subsystem mirroring method is used) data is a point-in-time image, and is consistent. Ensuring consistency requires an application that is written to recover in the event of a system crash. If the application does not support recovery, you must stop the application before initiating the FlashCopy. In this case, unmount the file systems before the FlashCopy to flush the file system cache to the file systems.

To create a FlashCopy of VGs used by a running application that supports recovery after a system crash, perform the following steps:

1. Put the application in a hot backup or quiesced mode if the application supports it. This speeds recovery of the application after it is started on the FlashCopy.
2. Use disk subsystem consistency groups for the disks in the VG.
3. Freeze the JFS2 file systems by using the `chfs` command if using Journaled File System 2 (JFS2). JFS2 is preferable because there is no similar function in JFS. For more information, see the `chfs` man page at:
   
4. Preferably, have one file system log per file system.
5. Initiate the FlashCopy.
6. Thaw the JFS2 file system.
7. Turn off the hot backup or quiesce mode of the application.
8. Use these procedures to get the new VG activated on the system:
   
   a. Run `logredo` against the file systems.
   b. Run `fsck` against the file systems.
9. Verify the consistency of the data using the application that uses that data.

For more information about ensuring file system and data consistency, see:


**4.5.2 Solaris and VxVM post-migration tasks**

This section addresses the steps required to access the migrated SUN data after the IBM Copy Services data migration is completed.

**Moving the application to the DS8000**

The steps for moving the application to the DS8000 are as follows:

1. Before you can split the Remote Mirror and Copy pairs, you need to stop any running application.
2. Unmount the file system using the `umount` command, and verify that the file system is unmounted using the `df` command as shown in Figure 4-44 on page 93.
Figure 4-44 Unmount the file system and verify the results of that operation

3. Stop the VxVM volume by entering `vxvol -g dg0 stop vol0`.

4. Deport the volume group by entering `vxdg deport dg0`.

5. Issue the `vxdisk offline IBM_SHARK0_0` command to VxVM offline the source LUN.

6. Verify the result of our activity by entering `vxdisk list`.

These last four steps are shown in Figure 4-45.

Figure 4-45 VxVM-specific operations to remove the source LUN from the control of VxVM
7. Look at the disks that are visible by Solaris by entering the `format` command as shown in Figure 4-46.

```bash
# format
Searching for disks...done

c2t500604843E0C4BCBd0: configured with capacity of 6.56MB

c4t500604843E0C4BD4d0: configured with capacity of 6.56MB

AVAILABLE DISK SELECTIONS:

0. c1t0d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
   /pci@8,600000/SUNW,qlc@4/fp@0,0/ssd@w21000004cfa37b97,0

1. c1t1d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
   /pci@8,600000/SUNW,qlc@4/fp@0,0/ssd@w21000004cfac3616,0

6. c2t50050763050C0786d4 <IBM-2107900-.437 cyl 4316 alt 2 hd 64 sec 256>
   /pci@8,600000/fibre-channel@1/fp@0,0/ssd@w50050763050c0786,4

7. c2t50050763050C0786d5 <IBM-2107900-.437 cyl 9205 alt 2 hd 30 sec 64>
   /pci@8,600000/fibre-channel@1/fp@0,0/ssd@w50050763050c0786,5

8. c2t50050763050C0786d6 <IBM-2107900-.437 cyl 9205 alt 2 hd 30 sec 64>
   /pci@8,600000/fibre-channel@1/fp@0,0/ssd@w50050763050c0786,6

9. c2t50050763050C39629d0 <IBM-2105800-.445 cyl 4316 alt 2 hd 64 sec 256>
   /pci@8,600000/fibre-channel@1/fp@0,0/ssd@w50050763050c39629,0

14. c4t50050763050C8786d4 <IBM-2107900-.437 cyl 4316 alt 2 hd 64 sec 256>
    /pci@8,700000/fibre-channel@1/fp@0,0/ssd@w50050763050c8786,4

15. c4t50050763050C8786d5 <IBM-2107900-.437 cyl 9205 alt 2 hd 30 sec 64>
    /pci@8,700000/fibre-channel@1/fp@0,0/ssd@w50050763050c8786,5

16. c4t50050763050C8786d6 <IBM-2107900-.437 cyl 9205 alt 2 hd 30 sec 64>
    /pci@8,700000/fibre-channel@1/fp@0,0/ssd@w50050763050c8786,6

17. c4t50050763050C9629d0 <IBM-2105800-.445 cyl 4316 alt 2 hd 64 sec 256>
    /pci@8,700000/fibre-channel@1/fp@0,0/ssd@w50050763050c9629,0

Specify disk (enter its number):
```

Figure 4-46 Using the format command to display the LUNs that are currently visible

Look for the source LUN paths `c4t5005076300CD9629d0s2` and `c2t5005076300C39629d0s2`.

8. Split the Remote Mirror and Copy mirror and enable the zones for the DS8000 on the switch so that the target LUN becomes visible.
9. Enter the `format` command as shown in Figure 4-47.

```
# format
Searching for disks...done

c2t500604843E0C4BCBd0: configured with capacity of 6.56MB
c4t500604843E0C4BD4d0: configured with capacity of 6.56MB

AVAILABLE DISK SELECTIONS:
  0. c1t0d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
      /pci@8,600000/SUNW,qlc@4/fp@0,0/ssd@w500604843E0C4BCBd0,0
  1. c1t1d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
      /pci@8,600000/SUNW,qlc@4/fp@0,0/ssd@w500604843E0C4BD4d0,0
  6. c2t50050763030B048Ed0 <IBM-2105800-4.45 cyl 4313 alt 2 hd 64 sec 256>
      /pci@8,600000/fibre-channel@1/fp@0,0/ssd@w50050763030B048Ed0,0
  7. c2t50050763050C0786d4 <IBM-2107900-.437 cyl 4316 alt 2 hd 64 sec 256>
      /pci@8,600000/fibre-channel@1/fp@0,0/ssd@w50050763050C0786d4,0
  8. c2t50050763050C0786d5 <IBM-2107900-.437 cyl 9205 alt 2 hd 30 sec 64>
      /pci@8,600000/fibre-channel@1/fp@0,0/ssd@w50050763050C0786d5,0
  9. c2t50050763050C0786d6 <IBM-2107900-.437 cyl 9205 alt 2 hd 30 sec 64>
      /pci@8,600000/fibre-channel@1/fp@0,0/ssd@w50050763050C0786d6,0
 10. c2t5005076300C39629d0 <IBM-2105800-4.45 cyl 4313 alt 2 hd 64 sec 256>
      /pci@8,600000/fibre-channel@1/fp@0,0/ssd@w5005076300C39629d0,0
 15. c4t50050763030B848Ed0 <IBM-2105800-4.45 cyl 4313 alt 2 hd 64 sec 256>
      /pci@8,700000/fibre-channel@1/fp@0,0/ssd@w50050763030B848Ed0,0
 16. c4t50050763050C8786d4 <IBM-2107900-.437 cyl 4316 alt 2 hd 64 sec 256>
      /pci@8,700000/fibre-channel@1/fp@0,0/ssd@w50050763050C8786d4,0
 17. c4t50050763050C8786d5 <IBM-2107900-.437 cyl 9205 alt 2 hd 30 sec 64>
      /pci@8,700000/fibre-channel@1/fp@0,0/ssd@w50050763050C8786d5,0
 18. c4t50050763050C8786d6 <IBM-2107900-.437 cyl 9205 alt 2 hd 30 sec 64>
      /pci@8,700000/fibre-channel@1/fp@0,0/ssd@w50050763050C8786d6,0
 19. c4t5005076300CD9629d0 <IBM-2105800-4.45 cyl 4313 alt 2 hd 64 sec 256>
      /pci@8,700000/fibre-channel@1/fp@0,0/ssd@w5005076300CD9629d0,0

Specify disk (enter its number):
```

**Figure 4-47 Using the format command to display the LUNs that are currently visible**

Comparing the output in Figure 4-46 on page 94 and Figure 4-47, you can see that you now have two more disks. Actually, they are actually two paths to the same target (the dual-pathed target device that you just made available to the operating system).

The output of the `format` command shows as an IBM 2105800. It is, however, a DS8000. The DS8000 is labeled incorrectly because the target LUN is an exact replica of the source Enterprise Storage Server LUN as created by Remote Mirror and Copy. The source Enterprise Storage Server LUN initially was correctly labeled as IBM 2105800 by Solaris, so the label (including the source geometry) was copied. However, do not rewrite the label on the target LUN because user data might be affected.
You can, however, show that the target disk is a DS8000 using the following steps:
a. Issue the `format` command as shown in Figure 4-48.

```
# format
Searching for disks...done

c2t500604843E0C4BCBd0: configured with capacity of 6.56MB
c4t500604843E0C4BD4d0: configured with capacity of 6.56MB

AVAILABLE DISK SELECTIONS:
0. c1t0d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
   /pci@8,600000/SUNW,qlc04/fp0,0/ssd0w21000004cfa37b97,0
1. c1t1d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
   /pci@8,600000/SUNW,qlc04/fp0,0/ssd0w21000004cfac3616,0
6. c2t50050763030B048Ed0 <IBM-2105800-4.45 cyl 4313 alt 2 hd 64 sec 256>
   /pci@8,600000/fibre-channel@1/fp0,0/ssd0w50050763050c0786,4
7. c2t5005076305C0786d4 <IBM-2107900-4.437 cyl 4316 alt 2 hd 64 sec 256>
   /pci@8,600000/fibre-channel@1/fp0,0/ssd0w50050763050c0786,5
8. c2t5005076305C0786d5 <IBM-2107900-4.437 cyl 9205 alt 2 hd 30 sec 64>
   /pci@8,600000/fibre-channel@1/fp0,0/ssd0w50050763050c0786,5
9. c2t5005076305C0786d6 <IBM-2107900-4.437 cyl 9205 alt 2 hd 30 sec 64>
   /pci@8,600000/fibre-channel@1/fp0,0/ssd0w50050763050c0786,6
10. c2t5005076300C39629d0 <IBM-2105800-4.45 cyl 4313 alt 2 hd 64 sec 256>
    /pci@8,600000/fibre-channel@1/fp0,0/ssd0w50050763050c39629,0
15. c4t50050763030B048Ed0 <IBM-2105800-4.45 cyl 4313 alt 2 hd 64 sec 256>
    /pci@8,700000/fibre-channel@1/fp0,0/ssd0w50050763050c8786,4
16. c4t5005076305C0786d4 <IBM-2107900-4.437 cyl 4316 alt 2 hd 64 sec 256>
    /pci@8,700000/fibre-channel@1/fp0,0/ssd0w50050763050c8786,5
17. c4t5005076305C0786d5 <IBM-2107900-4.437 cyl 9205 alt 2 hd 30 sec 64>
    /pci@8,700000/fibre-channel@1/fp0,0/ssd0w50050763050c8786,5
18. c4t5005076305C0786d6 <IBM-2107900-4.437 cyl 9205 alt 2 hd 30 sec 64>
    /pci@8,700000/fibre-channel@1/fp0,0/ssd0w50050763050c8786,6
19. c4t5005076300C9629d0 <IBM-2105800-4.45 cyl 4313 alt 2 hd 64 sec 256>
    /pci@8,700000/fibre-channel@1/fp0,0/ssd0w5005076300cd9629,0

Specify disk (enter its number): 6
```

Figure 4-48  Using the format command to list disks
b. Enter the number of the disk you want to investigate. The Format menu for that disk displays (Figure 4-49).

```
selecting c2t50050763030B048Ed0
[disk formatted]

FORMAT MENU:
- select a disk
- select (define) a disk type
- select (define) a partition table
- describe the current disk
- format and analyze the disk
- repair a defective sector
- write label to the disk
- surface analysis
- defect list management
- search for backup labels
- read and display labels
- save new disk/partition definitions
- show vendor, product and revision
- set 8-character volume name
- execute <cmd>, then return
- quit

format> inq
Vendor: IBM
Product: 2107900
Revision: .437
```

**Figure 4-49 Using the format menu to display inquiry data**

```
# vxdisk list

DEVICE      TYPE            DISK         GROUP        STATUS
EMC0_0      auto:none       -            -            online invalid
EMC0_1      auto:none       -            -            online invalid
EMC0_2      auto:none       -            -            online invalid
IBM_DS8x000_0 auto:none       -            -            online invalid
IBM_DS8x000_4 auto:none       -            -            online invalid
IBM_DS8x000_5 auto:none       -            -            online invalid
IBM_DS8x001_0 auto:cdsdisk    -            -            online udid_mismatch
IBM_SHARK0_1 auto             -            -            offline
c1t0d0s2    auto:none       -            -            online invalid
c1t1d0s2    auto:none       -            -            online invalid
```

**Figure 4-50 Using vxdisk list to show the status of our target device**

c. Enter `vxdisk list` to see the status of the target volume. Figure 4-50 shows that VxVM recognizes it as DS8000, despite the label on the disk that it is an IBM 2105-800. It also shows that it is attributed with a flag `udid_mismatch`. This flag shows that VxVM is aware that it is a cloned volume (Remote Mirror and Copy).
10. Import the disk group by entering `vxdg -o useclonedev=on -o updateid import dg0` (Figure 4-51).

```
# vxdg -o useclonedev=on -o updateid import dg0
# vxdisk list
DEVICE    TYPE            DISK         GROUP        STATUS
EMC0_0    auto:none       -            -            online invalid
EMC0_1    auto:none       -            -            online invalid
EMC0_2    auto:none       -            -            online invalid
IBM_DS8x000_0 auto:none    -            -            online invalid
IBM_DS8x000_4 auto:none    -            -            online invalid
IBM_DS8x000_5 auto:none    -            -            online invalid
IBM_DS8x001_0 auto:cdsdisk dg001        dg0          online clone_disk
IBM_SHARK0_1 auto            -            -            offline
c1t0d0s2  auto:none       -            -            online invalid
c1t1d0s2  auto:none       -            -            online invalid
```

*Figure 4-51  Importing the disk group*

11. Issue the `vxprint` command to see that the disk group was imported, but the VxVM volume itself is still disabled (Figure 4-52).

```
# vxprint
Disk group: dg0
TY NAME    ASSOC        KSTATE   LENGTH   PLOFFS   STATE    TUTILO  PUTILO
dg dg0      dg0          -        -        -        -        -       -

dm dg001    IBM_DS8x001_0 -       70598400 -        -        -       -

v vol0      fsugen        DISABLED 70596608 -        CLEAN  -       -
p1 vol0-01  vol0         DISABLED 70596608 -        CLEAN  -       -
sd dg001-01 vol0-01      ENABLED  70596608 0        ACTIVE -       -
```

*Figure 4-52  Entering vxprint to see volume status*

12. To be able to mount it later on, start the volume by entering `vxvol -g dg0 start vol0` as shown in Figure 4-53.

```
# vxvol -g dg0 start vol0
# vxprint
Disk group: dg0
TY NAME    ASSOC        KSTATE   LENGTH   PLOFFS   STATE    TUTILO  PUTILO
dg dg0      dg0          -        -        -        -        -       -

dm dg001    IBM_DS8x001_0 -       70598400 -        -        -       -

v vol0      fsugen        ENABLED  70596608 -        ACTIVE -       -
p1 vol0-01  vol0         ENABLED  70596608 -        ACTIVE -       -
sd dg001-01 vol0-01      ENABLED  70596608 0        ACTIVE -       -
```

*Figure 4-53  Starting the volume*
13. Mount the volume using the mount /ESS0 command (Figure 4-54).

```
# mount /ESS0
# df -k

Filesystem kbytes used avail capacity Mounted on
/dev/dsk/c1t0d0s0 33279289 5697589 27248908 18% /devices
/ctfs 0 0 0 0% /system/contract
/proc 0 0 0 0% /proc
/mnttab 0 0 0 0% /etc/mnttab
/sap 2731024 1296 2729728 1% /etc/svc/volatile
/objfs 0 0 0 0% /system/object
/platform/sun4u-us3/lib/libc_psr/libc_psr_hwcap1.so.1
33279289 5697589 27248908 18%
/platform/sun4u-us3/lib/panic.so.1
/platform/sun4u-us3/lib/sparcv9/libc_psr/libc_psr_hwcap1.so.1
33279289 5697589 27248908 18%
/platform/sun4u-us3/lib/sparcv9/lib_psr.so.1
/fd 0 0 0 0% /dev/fd
/swap 2729792 64 2729728 1% /tmp
/swap 2729776 48 2729728 1% /var/run
/swap 2729728 0 2729728 0% /dev/vx/dmp
/swap 2729728 0 2729728 0% /dev/vx/rdmp
/dev/vx/dsk/dg0/vol0 35298304 17205629 16961890 51% /ESS0
```

Figure 4-54 Mount the volume and verify that it became mounted

To verify the operation, check that the last entry in a file from the source LUN can also be seen on the target LUN (Figure 4-55).

```
# cd /ESS0
# tail -3 samplefile
Mar 26 17:14:06 SunFire280Rtic2 Corrupt label; wrong magic number
Mar 26 17:14:06 SunFire280Rtic2 scsi: [ID 107833 kern.warning] WARNING:
pci08,700
************** LAST ENTRY TO SAMPLEFILE ***************
```

Figure 4-55 Check data on target LUN

4.5.3 System z post-migration tasks

There are two scenarios to consider when completing the data migration: Migrating system DASD and migrating data. This section addresses at a high level the scenario of moving only data. The System z support personnel know best how to move to the new DS8000 storage unit. Use the following steps to move the data:

- Stop the applications.
- Delete the Remote Mirror and Copy pairs.
- Create new device entries in the input/output configuration data set (IOCDS) for the new DS8000.
- Activate the new IOCDS with the new DS8000 devices online at IPL and the Enterprise Storage Server devices offline at IPL. Also, ensure that future IPLs will use only DS8000 devices.
Stopping the applications
This step can be performed during a planned downtime of the system. All applications must be stopped so they can be moved to the new storage. This example assumes that software to automate the migration such as IBM GDPS® is not in place.

To stop the application successfully, perform the following steps:

1. Use the TSO command `tso cquery devin(e000) bitmap` as shown in Figure 4-56. The output shows that the application is still running and there are still out-of-sync tracks (168). The output also lists the four Remote Mirror and Copy paths available for this device. The source volume is E000 and the remote volume is E008. The current state of the pair is PENDING.XD.

   ![Figure 4-56 TSO query of global copy pair](image)

2. Stop the application.
3. Issue the `tso cquery devin(e000) bitmap` command again. The output shown in Figure 4-57 shows that the copy is now at 100%.

![Figure 4-57 Global copy pair status](image-url)
4. Issue the go-to-sync command and verify that the pair is in duplex state as shown in Figure 4-58.

CQUERY FORMATTED_LVL_4
VOLUME REPORT
************** PPRC REMOTE COPY CQUERY - VOLUME ***************
* (PRIMARY) (SECONDARY) *
* SSID CCA LSS SSID CCA LSS *
* DEVICE LEVEL STATE PATH STATUS SERIAL# SERIAL# *
* ------- --------- ----------  ----------- ---------    --------- *
* E000 PRIMARY.. DUPLEX.... ACTIVE.. E000 00 00 E008 00 08 *
* CRIT(NO)....... CGRPLB(YES) 0000000FNZT1 0000000FNZT1 *
* INCRES(NO). *
* PATHS PFCA SFCA STATUS: DESCRIPTION *
* ----- --------- ------  ------------------- *
* 4 0232 0302 13 PATH ESTABLISHED... *
* 0233 0303 13 PATH ESTABLISHED... *
* 0630 0700 13 PATH ESTABLISHED... *
* 0631 0701 13 PATH ESTABLISHED... *
* SUBSYSTEM WWNN LIC LEVEL *
* -----------  ----------------            -----------             *
* PRIMARY.... 5005076305FFFC6F3 5.2.420.266 *
***
* SECONDARY.1 5005076305FFC6F3 *
********************************************************************
CQUERY COMMAND COMPLETED FOR DEVICE E000. COMPLETION CODE: 00
***

Figure 4-58   Verify duplex state
Deleting the Remote Mirror and Copy pairs

For information about the TSO and ICKDSF commands to delete the Remote Mirror and Copy pairs, see 4.3.4, “Completing the data migration” on page 72. Using the TSO command `cquery` again shows that the volume is now in simplex state and all the paths are deleted (Figure 4-59).

```
CQUERY FORMATTED LVL 4
VOLUME REPORT
************** PPRC REMOTE COPY CQUERY - VOLUME **********************
*                                          (PRIMARY)    (SECONDARY) *
*                                          SSID CCA LSS SSID CCA LSS *
*DEVICE LEVEL STATE PATH STATUS SERIAL# SERIAL# *
*------ ---------  ----------  ----------- ---------    ---------  *
* E000 ........ SIMPLEX... INACTIVE E000 00 10 ............ *
* ............................................... 0000000FNZT1 ............ *
* ............................................... *
* PATHS SAID DEST STATUS: DESCRIPTION *
* ----- --------- ------  -------------------                      *
* 0 ---- ---- 00 NO PATH...........                      *
* ---- ---- 00 NO PATH...........                      *
* ---- ---- 00 NO PATH...........                      *
* ---- ---- 00 NO PATH...........                      *
* SUBSYSTEM WWNN LIC LEVEL *
* -------------- -------------                      *
* PRIMARY.... 5005076305FFC6F3 5.2.420.266             *
***
********************************************************************
CQUERY COMMAND COMPLETED FOR DEVICE E000. COMPLETION CODE: 00
***
```

Figure 4-59   Volume in simplex state and paths removed

Creating IOCDS entries

Perform this step in advance to decrease the downtime for the application. The system support personnel make the changes required on the system to recognize the new storage serial number and devices.

New DS8000 devices

Activate the new IOCDS so that the new DS8000 devices are online and the old storage devices offline at IPL. This change allows the system to use only the new DS8000 devices on any future IPLs.

The new DS8000 devices are varied online and available for use. The application is ready to restart.
DSCLbroker

The DSCLbroker is a scripting framework that automates Copy Services functions. The DSCLbroker provides the following features:

- Grouping volumes according to applications or other context
- Simplifies the execution of Copy Services commands
- Provides a scripting framework for implementing automation functions

This chapter contains the following sections:

- Overview
- System environment
- Programming techniques
- Automation techniques
- Considerations about configuration data
- Example migration using DSCLbroker
5.1 Overview

Besides the storage capacity for application data, the storage infrastructure of a modern data center must also replicate the data to other storage devices. With the Copy Services and interfaces like DSCLI, applications can use this replication by controlling Copy Services for their own purposes. These functions are commonly used for disaster prevention or data backup purposes.

Copy Services are frequently used for data migrations, but migrations are not a daily operation. Depending on your data center environment, applications, and type of migration, the migration tasks can get complicated. As a result, you might want to automate certain tasks, especially when the required actions cannot be accomplished using the standard storage management software.

The DSCLIBroker is a scripting framework that allows you to create user customized automation scripts. For example, consider a multitiered layered stack consisting of the DS8000 hardware as the lowest level and DSCLI above. You might want to write automation scripts where DSCLI commands are run against the storage. The DSCLIBroker can be positioned as an extra layer between the DSCLI and the applications (Figure 5-1).

![Figure 5-1 Positioning of DSCLIBroker](image)

The framework is written in Perl scripting language and consists of a series of perl library modules that support all DS8000 Copy Services functions. You can write your own scripts using the library. The framework also provides a perl script for each Copy Services DSCLI command that can be used without additional programming.

DSCLIBroker is available as an STG-Lab Service. Contact your IBM representative to order this service.

In the subject field, enter Migration using DSCLIBroker and continue to process the form.

If you are outside of IBM, talk to your IBM Representative about ordering the Service.

5.1.1 DSCLIBroker concepts

The complexity of Copy Service configuration increases as the number of applications involved and the size of those applications increases. The more applications must be migrated at the same time, the more care must be taken to map the correct volumes to the
applications. In addition, larger applications can have large lists of volumes that have copy relations to other volumes. Migrating data to the correct volumes is vital to avoid data inconsistency.

Using DSCLI Copy Services commands, with every command a list of copy relations must be specified. In a migration scenario with the steps: `mkpprc, lspprc, pausepprc, lspprc, failoverpprc` and again `lspprc`, the list of copy relations must be specified six times. It requires much effort to maintain these commands either in self-written scripts or on the DSCLI command line.

With DSCLibroker, the configuration data of the Copy Services relations is separated from the scripting code in a repository. In this repository, multiple Copy Services relations belonging to a single application can be grouped together and tagged with a name. When a DSCLibroker script is run, it refers to the tagged name, and the DSCLibroker then fetches all copy relations from the repository. Maintenance of these relationships is done in the repository, and so you do not need to change the scripting code.

In addition, the DSCLibroker provides a scripting framework that offers an easy way to write user-customized scripts. This framework is implemented in modular libraries written in Perl scripting language with an object-oriented approach. There is one library for each DS8000 Copy Services function available. The libraries themselves are organized so they can be extended to support other storage platforms too. The current storage platforms supported include DS8000, IBM DS6000™ and ESS Model 800. Plans are in place to support SAN Volume Controller and V7000 storage platforms and the TotalStorage Productivity Center for Replication command-line interface.

DSCLibroker libraries

The libraries are the core of the scripting framework. If you write Perl scripts using the DSCLibroker, you must include the libraries.

The following are the libraries and a short summary of their purposes and contents:

- **DSPPRC.pm**: This library is an object class where all remote copy functions are implemented. The functions include managing the paths and the pair relations.
- **DSFlashCopy.pm**: This library is an object class that holds all functions that maintain FlashCopy relations.
- **DSGlobalMirror.pm**: This library is an object class as well. It holds all functions related to DS8000 Global Mirror.
- **DSCLInator.pm**: This library is the meta class for the preceding classes. Commonly used functions for Copy Services are located here.
- **DSlib.pm**: This library contains global functions with no relations to Copy Services functions such as maintaining the DSCLibroker environment and querying and retrieving data from the repository.
- **DBbox.pm**: This library contains all necessary functions to communicate with the storage subsystems.
Figure 5-2 shows an overview of the libraries of the DSCLIBroker.

The data repository

All configuration data is stored in the data repository, and is used by the DSCLIBroker libraries when composing commands for the DSCLI. In this sense, configuration data includes the following information:

- Copy relations defined by source volume to target volume.
- Copy paths information for remote copy, which includes the relation source LSS to target LSS used by source port to target port.
- Definitions for DS8000 Global Mirror.
- Information about the storage systems.

Because some DSCLI commands use the same information, this information can be collected in the same repository entity. For example, when creating a Metro Mirror or a Global Copy, the same DSCLI command is used. Metro Mirror is denoted by the option -type mmir and Global Copy uses the option -type gcp. The remaining parameters are the source and target device, and the source and target volumes. A FlashCopy needs the same set of information except that the target device is not required. All these copy pair relations can be described with the same set of data, which is in the form of a database table or stanza file.
The complete set of required information is shown in Figure 5-3.

![Figure 5-3 DSCLbroker data model](image)

This set of tables is a normalized data model, which in theory can be used by a relational database system. For DSCLbroker however, the tables are implemented as flat stanza files because they can handle hundreds of thousands of entities without any problems. When the data migrations are done, the data in the repository is obsolete and can be discarded. In other engagements, the data in the repository must be maintained for a longer time. In these engagements, the data management capabilities of a database system might be more useful or even required.

As shown in Figure 5-3, each table has a name that is the key to a set of configuration data. A table might have an entry that refers to a key name of another table. For example, in the table CopyPairs.cfg, the entries for Source and Target reference a storage device defined in the table DSdev.cfg.

The following section includes an example for a complete repository definition where an application named SAPHRI manages a copy relation using the DSCLbroker. The whole setup includes the following steps:

1. Define the copy relations.
2. Define the storage devices.
3. Define the paths.
In the example, the application SAPHR1 is using the volumes 820A-820C and 8410-841F, which are two different contiguous volume ranges. The complete copy relation for this application is denoted in the stanza file CopyPairs.cfg. One stanza is created for the first volume range, and another for the second volume range. Both stanzas have the same name: SAPHR1_ab (Example 5-1).

Example 5-1  Defining a complex copy relation

CopyPair {
    name      = SAPHR1_ab
    type      = gcp
    srcss     = 82
    tgtss     = 8E
    srcvol    = 820A-820C
    tgtvol    = 8E0A-8E0C
    source    = ATS_3
    target    = ATS_1
    seqnum    =
    optset    = gcp_cascade
}

CopyPair {
    name      = SAPHR1_ab
    type      = gcp
    srcss     = 84
    tgtss     = 8E
    srcvol    = 8410-841F
    tgtvol    = 8E10-8E1F
    source    = ATS_3
    target    = ATS_1
    seqnum    =
    optset    = gcp_cascade
}

In some cases, one or more command-line options are required multiple times. For example, a Global Copy relation that is cascaded to an existing remote copy relation requires the option -cascade. This option can be placed in the repository using the stanza file OptSet.cfg as shown in Example 5-2. The defined option set gcp_cascade is referenced in the CopyPair stanza with the tag optset, as shown in Example 5-1.

Example 5-2  Define parameter in OptSet.cfg

OptSet {
    name    = gcp_cascade
    param   = cascade
    value   = yes
}

When this entry is referenced in a CopyPairs.cfg stanza, the option -cascade is the default every time DSCLibroker generates a DSCLI command for this relation.
In the CopyPair stanzas, the entries for **source** and **target** refer to the source and target storage devices. Both entries need to be defined in the `DSdev.cfg` file. In Example 5-3, the source device is named **ATS_3** and the target device is named **ATS_1**.

**Example 5-3  Definition of the storage devices**

```plaintext
DSdev {
    name = ATS_1
    IPaddress1 = 9.155.70.26
    IPaddress2 = 9.155.70.55
    WWNN = 5005076303FFC08F
    devID = IBM.2107-7503461
    DSCLIprof = script_03461.profile
    PWfile = script_03461.pwfile
    user = script
}

DSdev {
    name = ATS_3
    IPaddress1 = 9.155.62.97
    IPaddress2 = 9.155.62.97
    WWNN = 5005076303FFC1A5
    devID = IBM.2107-7520781
    DSCLIprof = script_20781.profile
    PWfile = script_20781.pwfile
    user = script
}
```

After identifying the storage devices, the paths must be specified. To establish the paths for copy relations, a physical link must be established first (see 4.4.2, “Creating Remote Mirror and Copy paths” on page 83). Verify that the links are available and the required information for the stanza entries can be obtained using the script `lsavailpprcport.pl`, as shown in Example 5-4.

**Example 5-4  Obtaining the port information for the paths**

```
$ lsavailpprcport.pl -source ATS_1 -target ATS_3 -srcIsp 84 -tgtIsp 8e
Local Port     Attached Port     Type
===================================================================
IBM.2107-7503461/I0141 IBM.2107-7520781/I0402 FCP
IBM.2107-7503461/I0142 IBM.2107-7520781/I0400 FCP
$ 
```

In this example, two ports have physical connections to the target storage device. Use this information to create the stanza file for the paths. For each repository stanza file, a script is

---

**Remember:** As you can see in the stanza, password files are used for the authentication to DSCLI. When writing scripts that runs command against the DSCLI, an automated authentication to the DSCLI is useful. Otherwise you must type in the user name and password each time the script is run. DSCLI offers a secure method to log in to the DSCLI automatically. For more information, see the *Command-Line Interface User’s Guide for the DS6000 series and DS8000 series*, GC53-1127, at:

available that generates the stanza information automatically. Use these scripts for the path definitions. The path definitions are the most vital in the repository because the configuration of the path can get complex. Only when the path definitions are correct will the subsequent copy commands work properly.

Use the gen_pprcpaths.pl command to generating path stanzas as shown in Example 5-5. The option -name SAPHR1 defines the base name of the stanza. The option -d 'f:ab' specifies that the paths data are created for the forward direction only. This option adds _ab to the stanza base name, which results in the real stanza name SAPHR1_ab. The option -l is used to specify the LSS relation as given. The option -p is used for the port pairs as shown by the lsavailpprc.pl command in Example 5-4 on page 111.

Example 5-5   Generating path stanzas

```bash
$ gen_pprcpaths.pl -name SAPHR1 -d 'f:ab' -l '82:8E 84:8E' -p 'I0141:I0402
I0142:I0400' -s ATS_1 -t ATS_3
```

```
##################################################
# PPRC paths for SAPHR1_ab
##################################################

#############
# LSS 82 -> 8E
# Box ATS_1 -> ATS_3
#############
PPRCpath {
  name = SAPHR1_ab
  srclss = 82
  tgtlss = 8E
  srcport = I0141
  tgtport = I0402
  Source = ATS_1
  Target = ATS_3
  consist = no
}

PPRCpath {
  name = SAPHR1_ab
  srclss = 82
  tgtlss = 8E
  srcport = I0142
  tgtport = I0400
  Source = ATS_1
  Target = ATS_3
  consist = no
}

#############
# LSS 84 -> 8E
# Box ATS_1 -> ATS_3
#############
PPRCpath {
  name = SAPHR1_ab
  srclss = 84
  tgtlss = 8E
  srcport = I0141
  tgtport = I0412
```


5.1.2 DSCLIbroker scripts

The DSCLIbroker comes with scripts that can be used as soon as the repository has the required configuration data. For FlashCopy, Global Copy, and Metro Mirror, the stanzas for the Storage device must contain the paths and the copy pairs. For more information, see 5.1.1, “DSCLIbroker concepts” on page 106. For Global Mirror, the stanzas GlobaMirror.cfg and Session.cfg must also be filled with data.

For each Copy Services-related DSCLI command, a corresponding DSCLIbroker script is available. Each script provides all command options that you can use in DSCLI. Some scripts have additional options that provide additional functionality. For example, using the -p option with the command failoverpprc.pl pauses the relation at the primary site before the failover. All scripts require the option -name, where the corresponding tagged name of the stanzas is requested. If no options are provided, you see a help text where the complete syntax is shown. Example 5-6 shows the help text for the command mkpprc.pl.

Example 5-6  Help output example

$ mkpprc.pl
Usage:
../mkpprc.pl

[-h|help]# This output
[-d|debug 1-4]# Set the debug level. Recommended level: 2
[-s|simulate]# Run script in simulate mode. Requires -debug 2
[-b|banner]# Prints out a banner
[-direction forward|reverse]
    # Specifies in which direction of the pairs
    # should operate. The default is -d forward.
[-t|type mmir|gcp]
    # Overrides the type specified in the stanzas
[-m|mode full|nocp]
    # Overrides the copy mode specified in the stanzas
[-cascade]
    # Enable relation to be cascaded
[-nocascade]# Overwrite the cascade option if set in OptSet
[-incresync enable|enableoinit|disable|recover|override]
    # apply incremental resync. Overrides the stanzas
[-to|tgtonline]# enables target to get online (z/OS only)
[-tr|tgtread]# Allows read access from target volumes
The following examples are based on the configuration data that was created in the previous chapter. You can now establish the Global Copy relation using the script `mkpprc.pl` as shown in Example 5-7. The only required parameter you must specify is the name of the stanza where all copy relations are defined. `SAPHRI1_ab` is the corresponding stanza as shown in Example 5-1 on page 110.

**Example 5-7 Establish the Global Copy with mkpprc.pl**

```bash
$ mkpprc.pl -n SAPHR1_ab
CMUC00153I mkpprc: Remote Mirror and Copy volume pair relationship 820A:8E0A successfully created.
CMUC00153I mkpprc: Remote Mirror and Copy volume pair relationship 820B:8E0B successfully created.
... snippet ...
CMUC00153I mkpprc: Remote Mirror and Copy volume pair relationship 841F:8E1F successfully created.
```

All other scripts work in the same way. The only required parameter is the name of the copy relation located in the stanzas. Additional parameters can be supplied depending on what you want to do. For an overview of the available options, use the option `-help` or run the command without any parameters. Example 5-8 shows how to pause the Global Copy.

**Example 5-8 Pause Global Copy with pausepprc.pl**

```bash
$ pausepprc.pl -n teamblack_ab
CMUC00157I pausepprc: Remote Mirror and Copy volume pair relationship 820A:8E0A successfully paused.
CMUC00157I pausepprc: Remote Mirror and Copy volume pair relationship 820B:8E0B successfully paused.
... snippet ...
CMUC00157I pausepprc: Remote Mirror and Copy volume pair relationship 841F:8E1F successfully paused.
```

In the previous examples, no information is provided about how the script is generating the DSCLI commands. To make the scripts more verbose, the `-d` (debug) option can be used. There are four levels of debug information available. The first debug level shows the generated DSCLI command.

Another helpful option is `-simulate`, which displays the DSCLI command but does not run it. This option can be used to verify whether the generated DSCLI command is the one you are expecting before it takes effect. The option `-d 1` also shows the generated command, but the command is run.

Example 5-9 shows the output when using the debug and simulate options.

**Example 5-9 Simulate mode and verbose command execution for mkpprc.pl**

```bash
$ mkpprc.pl -n SAPHR1_ab -simulate
```
When a PPRC relation has to be failed over, the DSCLI command `failoverpprc` must be issued at the target storage device. Provide the pair relations in the reverse order. The `failoverpprc.pl` script does both automatically. In Example 5-10 the generated `failoverpprc` command is displayed. Comparing it to the Example 5-9 on page 114, the device IDs of the storage devices and the pair relations are reversed.

**Example 5-10  Fail over Global Copy**

```
failoverpprc.pl -n SAPHR1_ab -d 1
Looking for name: SAPHR1_ab
  failoverpprc -dev IBM.2107-7503461 -remotedev IBM.2107-7503461 -type mmir 8E0A-8E0C:820A-820C
  8E10-8E1F:8410-841F
CMUC00196I failoverpprc: Remote Mirror and Copy pair 8E0A:820A successfully reversed.
CMUC00196I failoverpprc: Remote Mirror and Copy pair 8E0B:820B successfully reversed.
... snippet ...
CMUC00196I failoverpprc: Remote Mirror and Copy pair 8E1F:841F successfully reversed.
```

### 5.1.3 User customized scripts

You can write your own scripts if you have basic perl scripting language skills. The perl scripts must include the DSCLibbroker perl libraries as described in “DSCLibbroker libraries” on page 107. The DSCLibbroker libraries are designed as a modular set of building blocks to make creating scripts easier.

The following example demonstrates a simplified migration to target DS8000 storage devices using a Global Copy replication. The following steps are processed by the scripts:

- Create the paths to the target storage device.
- Establish a Global Copy and wait until all tracks have been copied.
- Pause the Global Copy.
- Fail over Global Copy to the target site.

Using this script, you can perform a series of migrations without any changes to the code. For each application that needs to be migrated, you must change the data in the corresponding stanza files.

Example 5-11 shows the complete script. This script uses the libraries `DSlib.pm`, where the debug levels and the simulation mode are included, and `DSPPRC.pm`, where all remote copy functions are defined. The libraries are referenced in lines 6 and 7 of the script.

**Example 5-11  Example migration script using DSCLibbroker**

```
1 #!/usr/bin/perl -w
2
```
3 #
4 # Include required library modules
5 #
6 use DSlib;
7 use DSPPRC;
8
9 #
10 # Declare command line options
11 #
12 use Getopt::Long;
13 my ($opt_debug,
14     $opt_simulate,
15     $opt_name,
16 );
17
18 GetOptions(
19     "d|debug:i" => \$opt_debug,   # --debug
20     "s|simulate" => \$opt_simulate,  # --simulate
21     "n|name=s" => \$opt_name,      # --name <stanza_name>
22 );
23
24 if (!defined $opt_name) {
25     print "option -name is required\n";
26     exit(3);
27 }
28
29 # set debug level and simulator mode
30 #
31 my $dbg_level=setDebugLevel($opt_debug);
32 my $dbg_simulate=setSimulateMode($opt_simulate);
33
34 # Define source and target device objects
35 #
36 my $source=DSbox->new();
37 my $target=DSbox->new();
38
39 # Create PPRC thingy
40 #
41 my $PPRC=DSPPRC->new($source,$target);
42
43 # Apply command for the primary site
44 #
45 $PPRC->mkpprcpath($opt_name);
46 $PPRC->mkpprc($opt_name,'forward','-wait');
47 $PPRC->pausepprc($opt_name);
48
49 # Execute primary DSCLI commands
50 #
51 my $ret=$source->DoDSCLIcommand('server');
In the lines 9 - 27 of the script, the required command options are declared. The **-name** option is required and therefore the lines 24 - 27 are checking whether this option is supplied. If the option is not given, an error message is reported and the script exits.

Verify the DSCLI commands that are generated by this script before they take effect. Therefore, the simulation mode is enabled when the command-line option **-simulate** is specified. The simulation mode is engaged in line 33.

In lines 35 - 44, the necessary objects are defined. There is one object for each storage device required, and an object that provides the remote copy function to the script.

In lines 49 - 51, the DSCLI commands for the paths and the copy pairs are generated. In this case, a Metro Mirror is established with the option **-wait**. The Metro Mirror allows all tracks to be copied to the secondary site before the operation continues. When the copying is complete, the copy relation is paused using the **pausepprc** command. This command sequence must be applied to the primary storage device.

In line 56, all DSCLI commands generated in the previous lines are sent to the primary storage device for execution.

In line 59, the failover to the secondary device is applied in the same manner. This command is sent to the auxiliary storage device. This script is now completed.

Example 5-12 shows the output of the simulation mode of the script, which is the generated sequence of DSCLI commands. The commands, up to the **pausepprc** command, are sent to the primary storage device. The **failoverpprc** command is sent to the target storage device.

**Example 5-12  Simulation mode output**

```
$ ITSOexamp.pl -name app1_ab -s
mkpprcpath -dev IBM.2107-75ABTV1 -remotedev IBM.2107-7503461 -remotewwnn
5005076303FFC08F -srclss 42 -tgtlss 64 -consistgrp I0011:I0142 I0012:I0141
mkpprcpath -dev IBM.2107-75ABTV1 -remotedev IBM.2107-7503461 -remotewwnn
5005076303FFC08F -srclss 53 -tgtlss 57 -consistgrp I0011:I0142 I0012:I0141
mkpprc -dev IBM.2107-75ABTV1 -remotedev IBM.2107-7503461 -mode full -type -wait
4200-4203:6400-6403 5300-5303:5700-5703
pausepprc -dev IBM.2107-75ABTV1 -remotedev IBM.2107-7503461 4200-4203:6400-6403
5300-5303:5700-5703
failoverpprc -dev IBM.2107-7503461 -remotedev IBM.2107-75ABTV1 -type mmir
6400-6403:4200-4203 5700-5703:5300-5303
$_
```
This script can be used for a series of migrations. In this example, the migrations are done application by application. In other words, this script will migrate one application after the other. For each migration, the same script is run, with the name of the application is supplied with the option -name. The only thing that must be changed are the copy relations for each application, which must be defined correctly in the CopyPairs.cfg stanza file.

5.1.4 Additional useful scripts

The DSCLIbroker framework includes additional scripts that help maintain the data repository. There are also other useful scripts that enable enhanced functions for the DSCLI commands. The following are the additional scripts and their functions:

- gen_dsdev.pl, gen_gmir.pl, gen_pprcpairs.pl, gen_pprcpaths.pl, gen_session.pl:
  These scripts generate the stanza files that correspond to their name. They format the stanza files and place the values you supply with the command file options in the correct places. The scripts gen_pprcpaths.pl and gen_pprcpairs.pl are especially useful because they allow you to list multiple relations in the command-line options. For more information about creating stanzas dynamically, see 5.4, “Automation techniques” on page 125.

- ValidatePairs.pl, ValidatePaths.pl:
  These scripts validate the pair and path configuration. This means that the content of the data in the stanza is compared against the real status on the storage devices.

- scanpprc.pl, scanflash.pl, scanpprcpath.pl, scansession.pl:
  The scan scripts obtain information from the storage devices and the output from the DSCLI is sent back to the DSCLIbroker as an XML data format. This information allows the scan commands to select specific parameters and display their values in the window.

- QUERY.pl:
  This script allows you to retrieve selected data from the repository. QUERY.pl searches after a pattern in a stanza file and prints out the results as a stanza or separated by commas.

- lspprc.pl -sum, lsflash.pl -sum:
  The option -sum with the scripts lspprc.pl and lsflash.pl prints the total number of Out-of-Sync tracks. This list is useful when many copy pairs are involved. Otherwise, to obtain the total amount of Out-of-Sync tracks for each single output line, the Out-of-Sync tracks must be summarized manually.

- lspprc.pl -waitnull, lsflash.pl -waitnull:
  The option -waitnull is similar to the -sum option, but the command will not return until all tracks have been copied to the remote target volumes.

5.2 System environment

The system environment of the DSCLIbroker is divided in a server part and a client part. In this section, the architecture of the DSCLIbroker, and how the communication and the maintenance are organized are described.

5.2.1 Architecture

The DSCLIbroker itself is organized as a client server application. The server part includes the broker itself. The broker is a daemon that waits for a request from a client to open a connection to a storage device. This connection is called a session. The session is opened by forking a child process where the DS8000 command-line interface (DSCLI) is started. The session waits at the input prompt. The broker manages the communication to the client.
The server where the DSCLI is running must have the role of a trusted storage management server. The server hardware must have access to the DS8000 storage systems, but they must also be accessible by storage administrators. This server can additionally be a gateway to the storage environment. Therefore it must be equipped with two different network interfaces. One interface is used for the administrator access and the second one is used to communicate with the storage environment. Figure 5-4 shows the system architecture.

The server also hosts the repository with all its stanza files. In this way, the configuration data is in one centralized spot and does not need to be distributed to the administrators.

![Figure 5-4 DSCLIbroker architecture](image)

The client part of the DSCLIbroker is dedicated to the storage administrators, who work with one or more storage devices. Every DSCLIbroker script that runs at the client site communicates with the server to perform the following tasks:

- Retrieve data from the repository
- Compose the DSCLI commands
- Send the commands back to the server

The server forwards the commands to the corresponding DSCLI daemon waiting in the background. When the command is run by the DSCLI, the results are sent back to the client.

### 5.2.2 Communication

The communication between the DSCLIbroker client and the server is organized in sessions. The storage administrator, working as a client, must establish a session to a storage device. Use the script `startSession.pl` to send a request with a name to the corresponding entry in
the DSdev.cfg stanza file to the broker. When the broker finds the stanza entry, it uses this information to fork a process call the *worker* and starts it in the DSCLI.

The broker sends the client a unique session ID and the port where the worker is listening. This information is located in a dedicated session file at the client. The broker then sends a `lssi` command to the DSCLI in the background. The output of this command is sent back to the client. Use this output to verify whether the session was opened against the correct DS8000 storage system.

A session can also be closed. The command `stopSession.pl` sends an appropriate request to the corresponding worker process that quits the DSCLI and terminates the process.

### 5.2.3 Users

Multiple users can be defined in DSCLI for each storage system. Users can also be defined using the DSCLIBroker. To do so, define an additional stanza in the DSdev.cfg stanza file for the same storage system. The user name must be declared (Example 5-3 on page 111). The session must be started with that user name supplied using the option `-user` as shown in Example 5-13.

**Example 5-13  Start session for a dedicated user.**

```
$ startSession.pl -box ATS_3 -user admin03
```

<table>
<thead>
<tr>
<th>Name</th>
<th>ID</th>
<th>Storage Unit</th>
<th>Model WWNN</th>
<th>State</th>
<th>ESSNet</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATS_20780</td>
<td>IBM.2107-7520781</td>
<td>IBM.2107-7520780</td>
<td>922 5005076303FFC1A5</td>
<td>Online</td>
<td>Enabled</td>
</tr>
</tbody>
</table>

dscli>

Session on port 55559 for ATS_3 with id 1311037799 was created successfully!

Although with DSCLI a user can log in multiple times to the same storage device, with DSCLIBroker only one session per storage device is allowed.

The authentication to the DSCLI is done using a password file, which must be generated manually in DSCLI. The password files must be located in the directory `.pwfiles` of the home directory of the DSCLIBroker scripting framework. This password file is an encrypted file, and access should be restricted to read access for the administrator only. The password file is generated with the DSCLI command `managepwfile`. See the DSCLI online help or the DSCLI documentation for details.

### 5.2.4 DSCLIBroker maintenance

The DSCLIBroker server is an autonomic process. However, there are a few maintenance tasks you need to perform.

The most important task is to start and stop the broker daemon using the script `Broker.pl start`. After the broker daemon is started, the DSCLIBroker server is ready to receive requests for starting sessions from any client. To clean up the DSCLIBroker server, run the same script with the parameter `stop`. This parameter stops all worker processes running in the background, and then stops the broker daemon itself.

To monitor the activities of the broker and the worker processes, a logging mechanism has been implemented. The log files are located in the directory `~/log` in the home directory of the
DSCLbroker installation. A numbered index is appended to the log file for each day. The log files are kept for 10 days before they are deleted.

5.2.5 Licenses

The DSCLbroker is licensed per function and number of storage devices that are managed by the DSCLbroker. For this reason, the serial numbers of the storage devices must be registered. There is no capacity-based license. For migrations, the DSCLbroker can be ordered as a tailored STG Lab Service, in which the DSCLbroker is available for no additional fee.

5.3 Programming techniques

The DSCLbroker scripting framework provides all functions of the DS8000 Copy Services, and some additional functions. However, the true function of DSCLbroker is to allow you to write your own code that takes advantage of the framework. Writing code is especially of interest for migrations, because certain operations cannot be covered by standard tools and software components provided with the storage devices.

5.3.1 Using DSCLbroker scripts

The DSCLbroker scripts are perl programs that can be used in shell scripts. These shell scripts are the easiest way to write scripts. With this method, you can implement interactions between applications or middleware components and the storage devices. The following are a few examples of what you can do with scripts:

- Stopping an application before taking a FlashCopy and then restarting the application afterward
- Alter a database into backup mode before failing over to the remote storage system
- Implement an interactive script that guides an administrator through a migration scenario

Example 5-14 shows a simple interactive migration script.

Example 5-14 Simple interactive migration script

```
1 #!/bin/ksh
2 #
3 4 echo "Validating the paths"
5 6 ValidatePaths.pl -n appl_ab
7 8 echo -n "Are all paths in correct state? [y/n]"
9 read yn
10 if [[ $yn = 'y' ]]
11 then
12     echo "Establish Global copy to replicate"
13     mkpprc.pl -n appl_ab -type gcp
14     lspprc.pl -n appl_ab -waitnull
15     echo "All tracks has been copied"
```
19     echo -n "Ready to fail over to remote [y/n]"
20     read yn
21     if [[ $yn = y ]]
22     then
23         failoverpprc.pl -n appl_ab -p
24     else
25         echo "Migration aborted"
26         exit 3
27     fi
28 else
29     echo "Please fix paths issues first"
30     exit 2
31 fi

In line 6 the paths are checked for the status and if all paths defined in the repository are available, and the results are displayed. In line 9, you are asked by the script to confirm the paths status before the script continues. If not, the script quits with a message. Otherwise the Global Copy is established in line 15 using the DSCLIbroker script `mkpprc.pl`. The synchronization of the Global Copy is monitored in line 16 using `lspprc.pl -waitnull`. With this option, the script waits until the synchronization completes. In line 19, you are asked if everything is ready to fail over. If not the script stops with a message. If you enter yes, the script `failoverpprc.pl` is applied.

You can use DSCLIbroker scripts in other script languages like python. In this case, system calls must be used that pass the command execution to the shell.

### 5.3.2 Using DSCLIbroker libraries

Using DSCLIbroker libraries is the most effective way to write user customized scripts. The libraries are written in the perl language, which means the program must also be written in perl. No special perl programming skills are required. However, the knowledge of object and complex data structures, and nested hashes can be helpful.

DSCLIbroker libraries provide additional functions that can be helpful to implement automated migrations tasks. DSCLIbroker libraries can be helpful in the following situations:

- To provide the correct information so the scripts make the correct decisions and implement a valid logic for the migration. The repository of the DSCLIbroker can contain this information, and the libraries provide functions to retrieve it from the repository.
- To obtain current information from the storage devices, for example to retrieve the status of paths or copy pairs.
- To run storage commands using the broker that are not related to copy services functions. For example, you can change the host connection to a target host after a successful replication of the data.

### Retrieving information from the repository

In this example, an operation in a migration scenario requires details of the Copy Service configuration. This information can be retrieved from the repository of the DSCLIbroker using the library function `getconfig` from the module `DSlib.pm` (Example 5-15).

```perl
Example 5-15   Retrieve information from the repository
use DSlib;
...```

122 Data Migration to IBM Disk Storage Systems
In this example, the source and the target volumes of a copy relation are retrieved. Include the DSlib library with the use statement somewhere at the top of your program. With getConfig, the CopyPairs.cfg stanza is queried for a certain relation as shown in the example. The result is an array of hashes, where each key of a hash element corresponds to a tag entry of the stanza. The value in each hash entry is the data you are looking for. To unpack this data, a simple for loop is used.

Getting information from the storage devices

Using the DSCLI, the usual way to obtain information about the storage configuration or status information is to use an appropriate list command such as lspprc. The default output format is a formatted table. However, when the script needs only specific values, this output format spends extra time parsing through the data. To make processing easier, the additional output formats stanza, delimiter, and XML are provided by the DSCLI.

The DSCLIBroker uses the XML output format from the DSCLI to parse for a required parameter. In the following example, the Out-of-Sync values for a copy relation are collected from the output of the lspprc.pl -sum command (Example 5-16).

Example 5-16 Parsing XML output from DSCLI

use DSPPRC;

# Create two new boxes because it is a remote copy
my $source=DSbox->new();
my $target=DSbox->new();

# Create PPRC thingy
my $PPRC=DSPPRC->new($source, $target);

# send 'lspprc -fmt xml' command using the pairs of 'myCopyRelation'
my @SCAN=$PPRC->scanpprc('myCopyRelation','','',-l');

# Collect the OOS from each pair
my $totalOOS=0;
for my $pair (@SCAN) {
    totalOOS += $pair->{outsynctrks};
}

print "Total number of OOS: $totalOOS\n";
The function `scanpprc` generates a `lspprc` command with the option `-l` for DSCLI, where the output format XML is requested. The XML format is transformed into an array of hashes where each hash represents the output of one pair relation from the `lspprc` output. To dissolve this structure and collect the Out-of-Sync tracks, a `for` loop searches through all hashes and lists the Out-of-Sync tracks as shown.

**Running storage commands**

The main focus of the DSCLIbroker functionality is the Copy Services functions for FlashCopy, Remote Copy, and Global Mirror, and combinations thereof. This means that the corresponding library modules are generating the DSCLI commands that are then sent to the broker. The broker then forwards the commands to the worker daemon with the DSCLI in the background.

However, the underlying libraries where the commands are sent to the broker can be used to issue any command. You can use them to allocate volumes, create volume groups, and so on.

Example 5-17 shows how to set a host connection for a target host system.

**Example 5-17 Making a host connection**

```
use DSbox;

# Create a box object
my $box=DSbox->new('ATS_1');

# Create DSCLI script
$box->WriteDSCLIscript('managehostconnect -volgrp v12 42');
$box->WriteDSCLIscript('lsconnect -volgrp v12 42');

# Execute DSCLI script
my $ret=$box->DoDSCLIcommand ('server');
exit($ret);
```

In this script, an object `$box` of the storage device `ATS_1` is allocated. All DSCLI commands are generated and run using this object. Any valid DSCLI command can be defined with the function `$box->WriteDSCLIscript`.

**Attention:** All DSCLI commands that are sent to the broker must not be interactive. When using removing commands such as `rmpprc` and `rmpprcpath`, make sure the option `-quiet` is used.

In this example, two DSCLI commands are collected into a script that is stored in the object. This DSCLI script is sent and executed to that storage device using the function `$box->DoDSCLIcommand`.

124  Data Migration to IBM Disk Storage Systems
5.4 Automation techniques

The DSCLIbroker provide a wide range of ways to automate migration tasks. Depending on the type of migration and the platforms and applications involved, you can reduce an entire migration down to a single operational task. This section covers the following automation techniques using DSCLIbroker:

- Subjects for writing automation scripts
- Generating DSCLI scripts
- Using control files

5.4.1 Subjects for writing automation scripts

The DSCLIbroker itself offers a certain level of automation, due to the organization of the data in the repository. The scripts included with the DSCLIbroker take advantage of this organization, but they mostly mimic DSCLI basic commands. Using the framework of the DSCLIbroker, automation can be more efficient, allowing a convenient and save migration operation.

Typically a migration consists of several steps of storage operations and operations that must be done on the servers. To determine where automation can be implemented, the technical workflow and the responsibilities must be defined first. Typically, you have a classification of organizational responsibilities in place, which is reflected in a dedicated storage management, server management and application management. Typically automation does not go across the management boundaries. A meaningful automation includes all steps that can be handled by a single operator. End the script at the hand over to the next system management instance in the workflow.

DSCLIbroker is typically assigned to the storage management organization. However, you can integrate the DSCLIbroker into other storage management automation tasks. For the storage management tasks, automation can be implemented for the following situations:

- Automating the storage migration
- Generating repository data

Automating the storage migration

The storage migration tasks are related to all tasks that must be applied against the storage subsystems. Tasks that can be run in a row can be implemented in a script as is. If you need to hand control over to another organization, you can have the automation script exit. Alternately, you can have the script wait at a user prompt until control comes back to the storage system management. The storage operator then confirms the return of the control to the script and it can continue with the next steps.

Generally implement simulation mode for the migration scenario so you can review the steps of the script before they take effect. During this process, print out detailed logging information and the DSCLI commands. In the simulation mode, no DSCLI commands are run. However, any commands that are display information can be issued and the output checked for accuracy before the real execution.

Generating repository data

It is mission critical for migration using the DSCLIbroker that the data in the repository is correct. Otherwise the correct volumes for a single application or a group of applications might not be copied to the correct target volumes. In large migration projects that take weeks or months, the production environment will probably change due to data growth or other reasons.
Typically, migration projects at that size are organized in waves, where groups of servers or applications are migrated in one go. Before starting a migration of such a wave, create the whole set of repository data for that wave. This guarantees that changes to the production environment are considered every time.

You can maintain the repository data by implementing a script that generates the data using the scripts mentioned in 5.1.4, “Additional useful scripts” on page 118. These scripts require a list of volumes that are mapped to the applications and the migration wave, which is applied next as input.

**Important:** You need to know which volumes are subject of the migration and which applications are affected before generating repository data.

### 5.4.2 Generating DSCLI scripts

In certain cases DSCLIbroker cannot be installed in your environment because of internal restrictions such as installation policies. In this case, you can use DSCLIbroker to support the migration by generating the DSCLI commands for each step using simulation mode (Example 5-12 on page 117). The output of the simulation mode of each DSCLIbroker script can then be used as a series of steps to complete the migration.

The DSCLIbroker provides a script that can generate these DSCLI scripts automatically. Therefore the repository data for the copy pairs, the storage devices, and optionally the paths must be available. When all the scenarios are defined, all DSCLI commands are extracted and collected in a control file. This control file is used to produce all DSCLI commands with the required parameters and volumes. Example 5-18 shows that a control file containing all commands is required for a Global Mirror migration.

**Example 5-18 Control file example that created all commands for Global Mirror**

<table>
<thead>
<tr>
<th>cmd</th>
<th>option</th>
<th>relation</th>
<th>outputfile</th>
</tr>
</thead>
<tbody>
<tr>
<td>mkpprc.pl</td>
<td>ab</td>
<td>LBG_mkpprc</td>
<td></td>
</tr>
<tr>
<td>mkflash.pl</td>
<td>bc</td>
<td>LBG_mkGMflash</td>
<td></td>
</tr>
<tr>
<td>mksession.pl</td>
<td>ab</td>
<td>LBG_mksession</td>
<td></td>
</tr>
<tr>
<td>pausepprc.pl</td>
<td>ab</td>
<td>LBG_pauseppc</td>
<td></td>
</tr>
<tr>
<td>failoverpprc.pl</td>
<td>ab</td>
<td>LBG_failoverpprc</td>
<td></td>
</tr>
<tr>
<td>reverseflash.pl</td>
<td>-fast -tgtpprc</td>
<td>bc</td>
<td>LBG_reverseflash</td>
</tr>
<tr>
<td>lspprc.pl</td>
<td>ab</td>
<td>LBG_lspprc</td>
<td></td>
</tr>
<tr>
<td>lspprc.pl</td>
<td>-r</td>
<td>ab</td>
<td>LBG_lspprc_remote</td>
</tr>
<tr>
<td>lspprc.pl</td>
<td>-l</td>
<td>ab</td>
<td>LBG_lspprc_long</td>
</tr>
<tr>
<td>lsflash.pl</td>
<td>bc</td>
<td>LBG_lsflash</td>
<td></td>
</tr>
<tr>
<td>lsflash.pl</td>
<td>-l</td>
<td>bc</td>
<td>LBG_lsflash_long</td>
</tr>
<tr>
<td>rmpprc.pl</td>
<td>ab</td>
<td>LBG_rmpprc</td>
<td></td>
</tr>
<tr>
<td>rmpprc.pl</td>
<td>-direction reverse</td>
<td>ab</td>
<td>LBG_rmpprc_remote</td>
</tr>
<tr>
<td>rmsession.pl</td>
<td>ab</td>
<td>LBG_rmsession</td>
<td></td>
</tr>
</tbody>
</table>

The DSCLI commands are written in text files into a certain directory. Each text file holds the commands for a single step of the scenario. Finally the DSCLI scripts themselves must be placed in the directory where the command files are located.
5.4.3 Using control files

Using control files with the DSCLIlbroker allows more automation. This automation can be used when you are performing multiple migrations and many different scenarios apply. Instead of writing a script or a program, control files are defined for each scenario.

In Example 5-18 on page 126, a control script was used to generate the DSCLI script. The format of the file used in this example is generally the same as for any other control file. The required parameters are the command itself, a list of parameters, and the stanza entry to which the operation applies. The other fields in the control file are optional, for example a name for a log file or a comment.

A single script can be used to read the content of a selected control file and run, step by step, the listed commands with their parameters. The control file can be selected using a command-line option. For each execution for a scenario, write logging information into a dedicated log file. Every DSCLIlbroker script has an option `-banner`, which allows you to supply comments that are printed at the top of each execution.

The advantage of this approach is that only one program must be developed in the beginning of the migration project. All migration scenarios are defined and maintained using the control files. No programming is required when a scenario must be changed.

5.5 Considerations about configuration data

This addresses considerations about the data that must be provided to the DSCLIlbroker. This data is configuration data of the copy relations of the volumes to be migrated. Use the following steps to provide the data to the DSCLIlbroker:

1. Define the data input format
2. Write a script to read in the configuration data
3. Generate the repository data using the appropriate DSCLIlbroker scripts

5.5.1 Data sources

For a migration using Copy Services, application data must be transmitted from a set of source volumes to corresponding target volumes. Typically, the target volumes are on a remote storage device. You need the volume information and the definition of the connectivity, including the communication ports, zoning, and path relations, to extract data for the DSCLIlbroker repository.

Usually you maintain this information using a data management system such as Tivoli TotalStorage Productivity Center for Replication, or even just a spreadsheet. In any case, a data export function that provides comma-separated value data is needed to import the data into the DSCLIlbroker repository.

The success of every migration depends on the quality of the data in the repositories. The information in the repository must represent the exact production environment. Use DSCLIlbroker as an interface for each migration. It is a vital connectivity between the production environment and the migration environment. Re-import the production data using this interface into the repository of the DSCLIlbroker before each migration.
5.5.2 Importing data to the repository

If the migration comprises many applications and servers, the whole migration project can take several weeks or even months. Normal production cannot be paused for that long, even on the applications that will be migrated. Therefore, the repository data might change due to capacity upgrades or any other configuration changes. The data in the repository of the DSCLibroker must be adopted accordingly before the migration can start.

Assuming you keep your configuration database at the most current state, import this data to the DSCLibroker repository just before the next migration is started. A script can read a csv export of your configuration data and generate a whole new set of the DSCLibroker repository automatically. Use the repository generation scripts as described in “Generating repository data” on page 125.

5.6 Example migration using DSCLibroker

This section shows an example of a real migration that was successfully accomplished for an important client. The example is shown according to the migration process as described in 2.3, “Data migration process” on page 18. The DSCLibroker is engaged during the analysis after the requirements for the needed automation are discovered.

5.6.1 Overview

The client was running several data centers in a metropolitan area. In a consolidation project, data center sites must be migrated to a new, larger site. In this context, two DS8000 storage devices and the connected servers must be moved to the new data center location. Both storage devices were the primary storage of a Global Mirror configuration. The new location was intended to be the new primary site, while the target site for the Global Mirror remains.

The whole migration comprised several thousand primary volumes and hundreds of applications. The server platforms were mostly AIX-based, but included VMware ESX server, OpenVMS, and zSeries server. The Global Mirror was managed using IBM TotalStorage Productivity Center for Replication. After the migration, a new TotalStorage Productivity Center for Replication server was established. The migration was organized in groups of applications that must be moved one by one to the new location.
Figure 5-5 shows the basic structure of the migration.

![Diagram]

Figure 5-5  Migration overview

Because of the huge number of applications, the projected duration of the whole project was several months. The main reason for this long time was the data center logistics like change management, hardware ordering, and installation management. Also, the client wanted the current production to be impacted as little as possible.

The general approach was to establish another Global Copy at the new data center by cascading the target volumes of the current Global Copy. During the initial copy phase, the production running at the original volumes was not affected. The volumes that had to be removed from the Global Mirror session and the direction of the new Global Copy reversed. The Target volumes and the FlashCopy relation at the secondary site were reused for the new Global Mirror. The new Global Mirror was started at the new data center site.

### 5.6.2 Analysis

In the analysis phase, the planned scenario had to be proofed for feasibility. The production environment was analyzed and dependencies discovered. In this case, the applications had a strong dependency on the server hardware because most applications where hosted in logical partitions LPARS that shared hardware. Therefore, the applications had to be migrated in groups according to their server hardware.

The original idea for the migration was to use TotalStorage Productivity Center for Replication to establish the cascaded Global Copy relation and to transfer that relationship to the new Global Mirror session. But it is not possible to establish the cascaded Global Copy relation to the new data center site without removing the current Global Mirror session. Therefore all migration steps were planned using DSCLI commands.
Given these two major topics, the analysis was that automation had to be implemented for two major reasons:

1. To move the correct set of volumes for each application
2. To avoid human error and guarantee consistency during command execution

DSCLIbroker provides a solution for both these concerns. Mapping applications to volumes is covered by supplying the repository with the correct data. In addition, the scripting framework was used to write scripts that fulfill the automation requirements.

### 5.6.3 Test and develop

The migration environment was rebuilt in an IBM lab. This test environment was used to develop and test the entire migration scenario. In addition to the scenarios for migration, scenarios to roll back to normal production if problems prevent migration from completing were.

Scripts to generate the repository data and the migration steps were developed and tested. For the repository data, the customer provided the TotalStorage Productivity Center for Replication session export. They also provided a spreadsheet with the applications, the volumes, and the group they were to be migrated with. Using both data sources, the repository data was generated for all entities.

The test environment was also used to educate the client about using the automation scripts and provide hands-on training.

### 5.6.4 Planning

With the tools and the scenarios developed in the testing phase, the final planning was finalized. All information was put in place and used to generate the instructions for every required production change management task.

A detailed time line plan was finalized after the test and development phase. All roles were assigned so the complete migration could be conducted by the customer.

### 5.6.5 Running

The migration execution was divided into two general steps.

A trial run a couple of days before the go-live migration was scheduled. In this trial run, the migration was run until the data was failed over from the storage perspective to the new data center. However, production was continued at the primary site. At the new data center, the applications were started using the migrated data and a health check was performed. After all tests were passed, the application was ready for the go-live migration.

In the second step, the go-live migration was run. This full migration ended when the production was started in the new data center.

### 5.6.6 Validating

Most validation was done during the trial run. This validation made sure that the data was replicated as expected, which was the final approval for the go-live migration.
Even with the go-live migration, the health checks were been issued as well to double check the success of the migration. During these validation steps, the data at the original production was preserved in case the client wanted to back out.
IBM XIV data migration

This chapter introduces the IBM XIV Storage System embedded data migration function, which is used to migrate data from a non-XIV storage system to the XIV Storage System. The XIV data migration function is included in the base XIV software, and is easy to deploy. This chapter includes usage examples and troubleshooting information.

At a high level, the steps to migrate to XIV using the XIV Data Migration function are:
1. Establish connectivity between the source device and XIV. The source storage device must have Fibre Channel or iSCSI connectivity with the XIV.

   **Important:** If the IP network includes firewalls between the mirrored XIV systems, TCP port 3260 must be open within the firewalls for iSCSI replication to work.

2. Identify in detail the configuration of the LUNs to be migrated.

3. Perform the data migration:
   - Stop and unconfigure all I/O from source-original LUNs.
   - Start data migration in XIV.
   - Map new LUNs to the host and discover new LUNs through XIV.
   - Start all I/O on the new XIV LUNs.

This chapter contains the following sections:
- Overview
- Handling I/O requests
- Data migration steps
- Command-line interface
- Manually creating the migration volume
- Changing and monitoring the progress of a migration
- Thick-to-thin migration
- Resizing the XIV volume after migration
- Troubleshooting
- Backing out of a data migration
- Migration checklist
- Other considerations
6.1 Overview

Whatever the reason for your data migration, you always want to avoid or minimize disruption to your business applications. While there are many options for migrating data, the XIV Storage System includes an embedded data migration feature. This feature allows you to easily migrate data from an existing storage system to the XIV Storage System. The production environment using the existing storage can continue functioning during the data transfer with only one brief period of downtime. Figure 6-1 illustrates a high-level view of what the data migration environment might look like.

The IBM XIV Data Migration solution offers a smooth data transfer for the following reasons:

- Requires only a single short outage to switch LUN ownership. This outage allows the immediate connection of a host server to the XIV Storage System. This connection provides you with direct access to all existing LUNs before they are copied to the XIV Storage System.

- Synchronizes data between the two storage systems using a method not apparent to the XIV Storage System as a background process with minimal performance impact.

- Supports data migration from practically all storage vendors.

- Can be used with Fibre Channel or iSCSI.

- Can be used to migrate SAN boot volumes.

The XIV Storage System manages the data migration by simulating host behavior. When connected to the storage device containing the source data, XIV looks and behaves like a SCSI initiator. In other words, it acts like a host server. After the connection is established, the storage device containing the source data acts as though it is receiving read or write requests from a host. In fact, the XIV Storage System is doing a block-by-block copy of the data, which the XIV then writes onto an XIV volume.
During the background copy process, the host server is connected to the XIV Storage System. The XIV Storage System handles all read and write requests from the host server, even if the data is not resident on the XIV Storage System. In other words, during the data migration, the data transfer is not apparent to the host and the data is available for immediate access.

The connections between the two storage systems must remain intact during the entire migration process. If at any time during the migration process the communication between the storage systems fails, the process also fails. In addition, if communication fails after the migration reaches synchronized status, writes from the host will fail if the source updating option was chosen. For more information, see 6.2, “Handling I/O requests” on page 135. The process of migrating data is performed at a volume level, as a background process.

The data migration facility in XIV firmware revisions 10.1 and later supports the following functions:

- Up to four migration targets can be configured on an XIV. A target is either one controller in an active/passive storage device or one active/active storage device. XIV firmware revision 10.2.2 increased the number of targets to 8. The target definitions are used for both Remote Mirroring and data migration. Both Remote Mirroring and data migration functions can be active at the same time. An active/passive storage device with two controllers can use two target definitions unless only one of the controllers is used for the migration.
- The XIV can communicate with host LUN IDs ranging from 0 to 512 (in decimal). This function does not necessarily mean that the non-XIV disk system can provide LUN IDs in that range. You might be restricted by the non-XIV storage controller to use only 16 or 256 LUN IDs (depending on hardware vendor and device).
- Up to 4000 LUNs can be concurrently migrated.

**Important:** The source system is called a target when setting up paths between the XIV Storage System and the non-XIV storage. This term is also used in Remote Mirroring, and both functions share terminology for setting up paths for transferring data.

## 6.2 Handling I/O requests

The XIV Storage System handles all I/O requests for the host server during the data migration process. Read requests are handled based on where the data is currently located. For example, if the data is already migrated to the XIV Storage System, it is read from that location. However, if the data is not yet migrated to the IBM XIV storage, the read request comes from the host to the XIV Storage System. The XIV System in turn retrieves the data from the source storage device.

### 6.2.1 Methods of handling write requests

The XIV Storage System handles all host server write requests, and the non-XIV disk system is not apparent to the host. All write requests are handled using one of two user-selectable methods, chosen when defining the data migration. The two methods are source updating and no source updating.
Selecting which method you want to use is shown in Figure 6-2. The check box must be selected to enable source updating, shown here as Keep Source Updated. If this box is not selected, changed data from write operations is only written to the XIV.

![Figure 6-2 Keep Source Updated check box](image)

**Source updating**

This method for handling write requests ensures that both storage systems are updated when a write I/O is issued to the LUN being migrated. Source updating keeps the source system updated during the migration process, and the two storage systems remain in sync after the background copy process completes. The write commands are only acknowledged by the XIV Storage System to the host after the following steps occur:

- Writing the new data to the local XIV volume
- Writing data to the source storage device
- Receiving an acknowledgement from the non-XIV storage device.

If there is a communication failure between the storage systems or a write fails, the XIV Storage System also fails the write operation to the host. This process ensures that the systems remain consistent. Application requirements will determine whether you use the Keep Source Updated option.

**No source updating**

This method for handling write requests ensures that only the XIV volume is updated when a write I/O is issued to the LUN being migrated. This method decreases the latency of write I/O operations because write requests are only written to the XIV volume. This limits your ability to back out a migration unless you have another way of recovering updates to the volume being migrated. You can avoid this risk by shutting down the host for the duration of the migration.

**Tip:** Do not select **Keep Source Updated** if migrating a boot LUN. Doing so allows you to quickly back out of a migration of the boot device if a failure occurs.
6.2.2 Multi-pathing with data migrations

There are essentially two types of enterprise storage systems when it comes to multi-pathing:

- **Active/active:** The volumes on these storage systems can be active on all of the storage system controllers at the same time. These systems support IO activity to any volume down two or more paths. These types of systems typically support load balancing capabilities between the paths with path failover and recovery in the event of a path failure. The XIV is such a device, and can use this technology during data migrations. Examples of IBM products that are active/active storage servers are the IBM DS6000, IBM DS8000, IBM ESS F20, IBM ESS 800, and IBM SAN Volume Controller. DS6000 and SAN Volume Controller are storage servers that have preferred controllers on a LUN-by-LUN basis. If attached hosts ignore this preference, there might be a small performance penalty.

  If your non-XIV disk system supports active/active, carefully configure multiple paths from XIV to non-XIV disk. The XIV load balances the migration traffic across those paths and automatically handles path failures.

- **Active/passive:** The volumes on these storage platforms can be active on only one controller at a time. These storage devices do not support I/O activity to any volume down multiple paths. Most support active volumes on one or more controllers at the same time, but a volume can be active on only one controller. An example of an IBM product that is an active/passive storage device is the IBM DS4700.

**Migrating from an active/active storage device**

If your non-XIV disk system supports active/active access, you can configure multiple paths from XIV to the non-XIV disk system. The XIV load balances the migration traffic across these paths. However, do not configure more than two connections or increase the initialization speed to a large value to speed up the migration. The XIV synchronizes one volume at a time per target, so with four targets, four volumes can be being migrated at once. Therefore, the speed of the migration is determined by the ability of the non-XIV storage device to read from the LUN being migrated. Unless the non-XIV storage device has striped the volume across multiple RAID arrays, the migration speed is unlikely to exceed 250–300 MBps. The speed is totally dependent on the non-XIV storage device.

**Important:** If multiple paths are created between an XIV and an active/active storage device, the same SCSI LUN IDs must be used for each LUN on each path. Otherwise, data corruption might occur. Configure a maximum of two paths per target because defining more paths does not increase throughput. With some storage arrays, defining more paths adds complexity and increases the likelihood of configuration issues and corruption.

**Migrating from an active/passive storage device**

Because of the active/active nature of XIV, special considerations must be made when migrating data from an active/passive storage device to XIV. Configure a single path between the non-XIV storage device controller and the XIV system. You might want to perform migrations with the host applications offline due to the single path.

Define the target to the XIV per non-XIV storage controller (controller, not port). Define at least one path from that controller to the XIV. All volumes active on the controller can be migrated using the defined target for that controller. For example, suppose that the non-XIV storage device contains two controllers (A and B):

- Define one target (Ctrl+A) with at least one path between the XIV and one controller on the non-XIV storage device (controller A). All volumes active on this controller can be migrated using this target. When defining the XIV initiator to the controller, define it as not supporting fail-over if that option is available on the non-XIV storage array. By doing so,
volumes that are passive on the \( A \) controller are not presented to the XIV. Check your non-XIV storage device documentation for more information.

- Define another target (Ctrl+B) with at least one path between the XIV and controller \( B \). All volumes active on controller \( B \) can be migrated to the XIV using this target. When defining the XIV initiator to the controller, be sure to define it as \textit{not} supporting failover if that option is available. By doing so, volumes that are passive on controller \( B \) are not presented to the XIV. Check your non-XIV storage device documentation for more information.

shows the

*Figure 6-3 Active/Passive as multiple targets*

**Tip:** If your controller has two target ports (DS4700, for example), both can be defined as links for that controller target. Make sure that the two target links are connected to separate XIV modules. Connecting the links this way makes one redundant in a module failure.

**Important:** Certain examples shown in this chapter are from an IBM DS4000® active/passive migration with each DS4000 controller defined independently as a target to the XIV Storage System. If you define a DS4000 controller as a target, do not define the alternative controller as a second port on the first target. Doing so causes unexpected issues such as migration failure, preferred path errors on the DS4000, or slow migration progress.

### 6.3 Data migration steps

The following are the high-level steps required when migrating a volume from a non-XIV system to the XIV Storage System:

1. Set up the initial connection:
   - Zone or cable the XIV to the non-XIV storage device.
   - Define XIV to a non-XIV storage device (as a host).
   - Define non-XIV storage device to XIV (as a migration device).

2. Create a data migration volume on XIV:
   - Perform pre-migration tasks for the host being migrated:
     - Back up your data.
     - Shut down your host or application, or unmount the file system.
     - Perform a point-in-time copy of original non-XIV volume if possible.
     - Unzone the host from non-XIV storage.
   - Define and test the data migration volume:
• On non-XIV storage, map volumes away from host and map them instead to XIV.
• On XIV, create data migration and test it.

3. Activate data migration on XIV.

4. Define the host on XIV and bring the host online:
   – Zone the host to XIV.
   – On XIV, map volumes to the host.
   – Bring the host online:
     • Update host HBA drivers and firmware.
     • Install the Host Attachment Kit and detect volumes.

5. Complete the data migration:
   – Monitor the migration on XIV.
   – Delete the migration on XIV.

These steps are explained in detail in the following sections.

6.3.1 Initial connection setup

For the initial connection setup, zone or cable the XIV to the system being migrated.

**Zoning or cabling the XIV to the non-XIV storage device**

To connect the two devices, perform the following steps:

1. Run cables from port 4 on each selected XIV interface module to a fabric switch.

2. Zoned the XIV initiator ports (whose WWPNs end in 3) to the selected non-XIV storage device host ports using single initiator zoning. Each zone contains one initiator port and one target port.

Because the non-XIV storage device views the XIV as a host, the XIV must connect to the non-XIV storage system as a SCSI initiator. Therefore, the physical connection from the XIV must be from initiator ports on the XIV. The default initiator port for Fibre Channel is port 4 on each active interface module. The initiator ports on the XIV must be fabric attached, so they need to be zoned to the non-XIV storage system.

Use two physical connections from two separate modules on two separate fabrics for redundancy. However, redundant pathing is not possible on active/passive controllers.

It is possible that the host might be attached through one medium (such as iSCSI), whereas the migration occurs through the other. In this case, the host-to-XIV connection method and the data migration connection method are independent of each other.

Depending on the non-XIV storage device vendor and device, it might be easier to zone the XIV to the ports where the volumes being migrated are already present. Zoning in this way might avoid needing to reconfigure the non-XIV storage device. For example, in EMC Symmetrix/DMX environments, it is easier to zone the fiber adapters (FAs) to the XIV where the volumes are already mapped.
Figure 6-4 depicts a fabric-attached configuration. It shows that module 4 port 4 is zoned to a port on the non-XIV storage through Fabric A. Module 7 port 4 is zoned to a port on the non-XIV storage through Fabric B.

Defining XIV to the non-XIV storage device as a host

After the physical connection between the storage devices is complete, the XIV initiator (WWPN) must be defined on the non-XIV storage device. This process is vendor- and device-dependent because you must use the non-XIV storage device management interface. See the documentation for the non-XIV storage device for information about how to configure hosts. The XIV is seen as a host to the non-XIV storage.

If you already zoned the XIV to the non-XIV storage device, the WWPNs of the XIV initiator ports might be displayed in the WWPN list. Whether they are displayed depends on the non-XIV storage device and storage management software. If they are not there, you must manually add them. The WWPNs might not be displayed if you need to map an LUN0 or if SAN zoning is not done correctly.

The XIV must be defined as a Linux or Windows host to the non-XIV storage device. If the non-XIV device offers several variants of Linux, you can select SuSE or RedHat Linux, or Linux x86. Selecting the host defines the correct SCSI protocol flags for communication between the XIV and non-XIV storage device. The principal criterion is that the host type must start LUN numbering with LUN ID 0. If the non-XIV storage device is active/passive, check whether the host type selected affects LUN failover between controllers. For more information, see 6.12, “Other considerations” on page 173.

There might also be other vendor-dependent settings. For more information, see 6.12, “Other considerations” on page 173.
Define non-XIV storage device to XIV (as a migration target)

After the physical connectivity is made and the XIV is defined to the non-XIV storage device, the non-XIV storage device must be defined on the XIV. This process includes defining the following aspects:

- The storage device object
- The WWPN ports on the non-XIV storage device
- The connectivity between the XIV and the non-XIV storage device.

To complete this process, perform the following steps:

1. In the XIV GUI, click Remote → Migration Connectivity.
2. Click Create Target as shown in Figure 6-5.

   **Tip:** If Create Target is disabled, you have reached the maximum number of targets. The number of allowed targets includes both migration and mirror targets.

3. Make the appropriate entries and selections, then click Define (Figure 6-6).
   - Target Name: Enter a name of your choice.
   - Target Protocol: Select FC from the list.
4. Click the gray line to access the migration connectivity (Figure 6-7).

**Tip:** The data migration target is represented by an image of a generic rack. If you want to delete or rename the migration device, right-click that rack.

![Figure 6-7 Click the gray line](image1)

5. Right-click the dark box that is part of the defined target and select **Add Port** (Figure 6-8).

![Figure 6-8 Defining the target port](image2)

6. Enter the WWPN of the first (fabric A) port on the non-XIV storage device zoned to the XIV. There is no list, so you must manually type or paste in the correct WWPN.

**Tip:** You do not need to use colons to separate every second number. It makes no difference if you enter a WWPN as 10:00:00:c9:12:34:56:78 or 100000c912345678.

7. Click **Add**.
8. Enter another port (repeating step 3) for those storage devices that support active/active multi-pathing. This port can be the WWPN that is zoned to the XIV on a separate fabric.

9. Connect the XIV and non-XIV storage ports by clicking and dragging from the port on the XIV to the port (WWPN) on the non-XIV storage device (Figure 6-9). In the example, the mouse started at module 9 port 4 and has nearly reached the target port. The connection turns red when the line reaches port 1 on the target.

![Figure 6-9 Dragging a connection between XIV and migration target](image-url)
Figure 6-10 shows the active connection from module 9 port 4 to port 1 on the non-XIV storage device as a green line. This means that the non-XIV storage system and XIV are connected and communicating, which includes the following concerns:

- SAN zoning is done correctly.
- The correct XIV initiator port is selected.
- The correct target WWPN is entered and selected.
- LUN 0 was detected on the target device.

If there is an issue with the path, the connection line is red.

Tip: Ensuring that LUN0 is visible on the non-XIV storage device down the controller path that you are defining helps ensure functional connectivity. Connections from XIV to DS4000, EMC DMX, or Hitachi HDS devices require a real disk device to be mapped as LUN0. However, other devices, such as IBM ESS 800 and EMC CLARiiON, do not need a LUN to be allocated to the XIV.

6.3.2 Creating a data migration volume on XIV

This section addresses the following steps needed to create a data migration volume on XIV:

- Performing pre-migration tasks for the host being migrated
- Defining and testing the data migration volume
Performing pre-migration tasks for the host being migrated

Follow these steps to prepare the host:

1. Back up the volumes being migrated.

   A full restorable backup must be created before any data migration activity. Verify the backup to make sure all the data is restorable and that there are no backup media errors.

2. Shut down the application/host.

   Before the actual migration can begin, the application must be quiesced. Stopping the I/O ensures that the application data is in a consistent state. Because the host might need to be rebooted a number of times before the application data being available again, consider the following steps:
   
   – Set applications to not automatically start when the host operating system restarts.
   – Stop file systems from being automatically remounted on boot. For operating systems based on UNIX, comment out all affected file system mount points in the fstab or vfstab.

   **Note:** In clustered environments, you might choose to work with only one node until the migration is complete. If so, consider shutting down all other nodes in the cluster.

3. Perform a point-in-time copy of the volume on the non-XIV storage device if that function is available. Perform the copy before changing any host drivers or installing new host software, particularly if you are going to migrate boot from SAN volumes.

4. Unzone the host from non-XIV storage. The host must no longer access the non-XIV storage system after the data migration is activated. The host must perform all I/O through the XIV.

Defining and testing the data migration volume

Perform these steps to define and test the data migration volume:

1. Allocate the non-XIV volume to XIV.

   The volumes being migrated to the XIV must be allocated using LUN mapping to the XIV. The LUN ID presented to the XIV must be a decimal value from 0 to 512. If using hexadecimal LUN numbers, the LUN IDs can range from 0x0 to 0x200. These LUN IDs must be converted to decimal when entered into the XIV GUI. The XIV does not recognize a host LUN ID above 512 (decimal). Figure 6-11 shows LUN mapping using a DS4700. It depicts the XIV as a host called XIV_Migration_Host with four DS4700 logical drives mapped to the XIV as LUN IDs 0 - 3.

![Figure 6-11 Non-XIV LUNs defined to XIV](image_url)
When mapping volumes to the XIV it is important to note the LUN IDs allocated by the non-XIV storage. The methodology to do assign LUN IDs varies by vendor and device, and is documented in greater detail in 6.12, “Other considerations” on page 173.

**Important:** You must unmap the volumes from the host during this step, even if you plan to power the host off during the migration. The non-XIV storage presents only the migration LUNs to the XIV. Do not allow the host to detect the LUNs from both the XIV and the non-XIV storage.

2. Define data migration object/volume.

After the volume being migrated to the XIV is allocated to the XIV, a new data migration volume can be defined. The source volume from the non-XIV storage system and the XIV volume must be the same size. It is easiest to allow XIV create the target LUN for you as explained to the following section. If you want to manually create the volumes on the XIV, see 6.5, “Manually creating the migration volume” on page 157.

**Important:** You cannot use the XIV data migration function to migrate data to a source volume in an XIV remote mirror pair. If you need to use a remote mirror pair, migrate the data first and then create the remote mirror after the migration is completed.

**Creating an XIV volume automatically**

The XIV can determine the size of the non-XIV volume and create the XIV volume when the data migration object is defined. This simple method avoids potential problems when manually calculating the real block size of a volume. To create an XIV volume automatically, perform the following steps:

1. In the XIV GUI, click Remote → Migration.
2. Right-click Migration and select Define Data Migration.
3. In the Define Data Migration window (Figure 6-12 on page 147), make the appropriate selections and entries:
   - **Destination Pool:** Select the pool where the volume will be created.
   - **Destination Name:** Enter a user-defined name that for the local XIV volume.
   - **Source Target System:** Select the already defined non-XIV storage device.

   **Important:** If the non-XIV device is active/passive, the source target system must represent the controller (or service processor) on the non-XIV device that currently owns the source LUN. Find out from the non-XIV storage which controller is presenting the LUN to the XIV.

   - **Source LUN:** Enter the decimal value of the host LUN ID as presented to the XIV from the non-XIV storage system. Certain storage devices present the LUN ID as hex. The number in this field must be the decimal equivalent. Ensure that you do not accidentally use internal identifiers that you might also see on the source storage systems management windows. In Figure 6-11 on page 145, the correct values are in the LUN column.
Keep Source Updated: Select this check box if the non-XIV storage system source volume is to be updated with writes from the host. If you select this option, all writes from the host are written to both volumes until the data migration object is deleted.

Note: Do not select Keep Source Updated if you are migrating the boot LUN so that you can quickly back out if a failure occurs.

Figure 6-12 Define Data Migration object/volume

4. Click Define.

The migration begins as shown in Figure 6-13. Define Data Migration queries the configuration of the non-XIV storage system and create an equal sized volume on XIV. To check whether you can read from the non-XIV source volume, run Test Data Migration.

Figure 6-13 Defined data migration object/volume

Consideration: On some active/passive non-XIV storage systems, the configuration can be read over the passive controller, but Test Data Migration fails.
5. Right-click the created data migration object and select **Test Data Migration**. If there are any issues with the data migration object the test fails and the issues encountered are reported (Figure 6-14).

![Figure 6-14  Test Data Migration option](image)

**Tip:** If you are migrating volumes from a Microsoft Cluster Server (MSCS) that is still active, migration testing might fail due to reservations on the source LUN placed by MSCS. You must bring the cluster down properly to get the test to succeed. If the cluster is not brought down properly, errors occur either during the test or when activated. The SCSI reservation must then be cleared for the migration to succeed.

### 6.3.3 Activate a data migration on XIV

After the data migration volume is tested, you can begin the actual data migration. When data migration is initiated, the data is copied sequentially in the background from the non-XIV storage system volume to the XIV. The host reads and writes data to the XIV storage system without being aware of the background I/O being performed.

**Important:** After it is activated, the data migration can be deactivated. However, after deactivating the data migration, the host is no longer able to read or write to the migration volume and all host I/O stops. Do not deactivate the migration with host I/O running. If you want to abandon the data migration, see the back-out process described in section 6.10, “Backing out of a data migration” on page 169.

Right-click to select the data migration object/volume and choose **Activate**. Activate all volumes being migrated so that they can be accessed by the host. The host has read and write access to all volumes, but the background copy occurs serially volume by volume. If two targets (such as non-XIV1 and non-XIV2) are defined with four volumes each, two volumes are actively copied in the background: One volume from non-XIV1 and another from non-XIV2. All eight volumes are accessible by the hosts.
Figure 6-15 shows the menu choices when right-clicking the data migration. The Test Data Migration, Delete Data Migration, and Activate menu items are the most-used commands.

<table>
<thead>
<tr>
<th>Name</th>
<th>Size (GB)</th>
<th>Status</th>
<th>Remote LUN</th>
<th>Remote System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows2003_D</td>
<td>53</td>
<td>Inactive</td>
<td>0</td>
<td>DS4700:Ctrl-B</td>
</tr>
<tr>
<td>Windows2003_E</td>
<td>10</td>
<td>Inactive</td>
<td>1</td>
<td>DS4700:Ctrl-B</td>
</tr>
<tr>
<td>Windows2003_F</td>
<td>74</td>
<td>Inactive</td>
<td>2</td>
<td>DS4700:Ctrl-B</td>
</tr>
<tr>
<td>Windows2003_G</td>
<td>53</td>
<td>Inactive</td>
<td>3</td>
<td>DS4700:Ctrl-B</td>
</tr>
</tbody>
</table>

Figure 6-15  Activate data migration

6.3.4 Defining the host on XIV and bringing the host online

Defining the host on the XIV and bringing to online involves the following steps:

- Zoning the host to XIV
- Defining the host being migrated to the XIV
- Mapping volumes to the host on XIV
- Bringing the host online

Zoning the host to XIV

The host must first be directed using SAN fabric zoning to the XIV instead of the non-XIV storage system. It must be directed because the XIV is acting as a proxy between the host and the non-XIV storage system. The host must no longer access the non-XIV storage system after the data migration is activated. The host must perform all I/O through the XIV.

Defining the host being migrated to the XIV

Before performing data migrations and allocating the volumes to the hosts, the host must be defined on the XIV. Volumes are then mapped to the hosts or clusters. If the host is to be a member of a cluster, the cluster must be defined first. However, a host can be easily moved from or added to a cluster at any time. Moving the host requires that the host is zoned to your XIV target ports through the SAN fabric.

Perform the following steps to define the host:

1. To define a cluster (optional):
   a. In the XIV GUI, click Host and Clusters → Host and Clusters.
   b. Click Add Cluster.
   c. Enter a cluster name in the Name field.
   d. Click OK.
2. To define a host:
   a. In the XIV GUI, click Host and Clusters → Host and Clusters.
   b. Click Add Host from the top menu bar. Make the appropriate entries and selections.
      i. Name: Enter a host name.
      ii. Cluster: If the host is part of a cluster, select the cluster from the list.
c. Click **Add**.

d. Right-click the host and select **Add Port**.

e. Enter the following information:

   i. **Port Type**: Select **FC** from the list.

   ii. **Port Name**: This option produces a list of WWPNs that are logged in to the XIV, but that are not assigned to a host. WWPNs can be chosen from the list or entered manually.

   iii. Click **Add**.

f. Repeat these steps to add all the HBAs of the host being defined.

### Mapping volumes to the host on XIV

After the data migration is started, you can use the XIV GUI or XCLI to map the migration volumes to the host. When mapping volumes to hosts on the XIV, LUN ID 0 is reserved for XIV in-band communication. This means that the first LUN ID that you normally use is LUN ID 1. This restriction includes boot-from-SAN hosts. You might also choose to use the same LUN IDs as were used on the non-XIV storage, but this is not mandatory.

**Important**: The host cannot read the data on the non-XIV volume until data migration is activated. The XIV does not pass through (proxy) I/O for an inactive migration. If you use the XCLI `dm_list` command to display the migrations, ensure that Yes is shown in the **Active** column for every migration.

### Bringing the host online

After the volumes are mapped to the host server, the host can be brought online.

The host must be configured using the XIV host attachment procedures. These procedures include the following steps:

- Removing any existing/non-XIV multi-pathing software
- Installing the native multi-pathing drivers and recommended patches,
- Installing the XIV Host attachment kit, as identified in the **XIV Host Attachment Guides**
- Installing the most current HBA driver and firmware

One or more reboots might be required. Documentation and software can be found at: [http://www.ibm.com/support/search.wss?q=ssgl*&tc=STJTAG+HW3E0&rs=1319&dc=D400&dtm](http://www.ibm.com/support/search.wss?q=ssgl*&tc=STJTAG+HW3E0&rs=1319&dc=D400&dtm)

When volume visibility is verified, the application can be brought up and operations verified.

**Tip**: In clustered environments, bring only one node of the cluster online initially after the migration is started. Leave all other nodes offline until the migration is complete. After the migration is complete, update all other nodes (driver, host attachment package, and so on), in the same way as the primary node. For more information, see “Performing pre-migration tasks for the host being migrated” on page 145.

### 6.3.5 Completing the data migration on XIV

To complete the data migration, perform the following steps:

- Tracking the data migration progress
- Deleting data migration
Tracking the data migration progress

Figure 6-16 shows the progress of the data migrations. The status bar can be toggled between GB remaining, percent complete, and hours/minutes remaining. The example shows four data migrations, one of which has started background copy and three of which have not. Only one migration is being copied at this same time because there is only one target (DS4700_Ctrl_B).

<table>
<thead>
<tr>
<th>Name</th>
<th>Size (GB)</th>
<th>Status</th>
<th>Remote LUN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows2003_D</td>
<td>53</td>
<td>Initialization (3%)</td>
<td>DS4700_Ctrl_B</td>
</tr>
<tr>
<td>Windows2003_E</td>
<td>10</td>
<td>Initialization (6%)</td>
<td>DS4700_Ctrl_B</td>
</tr>
<tr>
<td>Windows2003_F</td>
<td>74</td>
<td>Initialization (0%)</td>
<td>DS4700_Ctrl_B</td>
</tr>
<tr>
<td>Windows2003_G</td>
<td>53</td>
<td>Initialization (0%)</td>
<td>DS4700_Ctrl_B</td>
</tr>
</tbody>
</table>

Figure 6-16 Data migration progress

After all of the data in a volume is copied, the data migration achieves synchronization status. After synchronization is achieved, all read requests are served by the XIV Storage System. If source updating was selected, the XIV continues to write data to both itself and the outgoing storage system until the data migration is deleted. Figure 6-17 shows a completed migration.

<table>
<thead>
<tr>
<th>Name</th>
<th>Size (GB)</th>
<th>Status</th>
<th>Remote LUN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows2003_D</td>
<td>53</td>
<td>Synchronized</td>
<td>0</td>
</tr>
<tr>
<td>Windows2003_E</td>
<td>10</td>
<td>Synchronized</td>
<td>1</td>
</tr>
<tr>
<td>Windows2003_F</td>
<td>74</td>
<td>Synchronized</td>
<td>2</td>
</tr>
<tr>
<td>Windows2003_G</td>
<td>53</td>
<td>Synchronized</td>
<td>3</td>
</tr>
</tbody>
</table>

Figure 6-17 Data migration complete

Deleting data migration

After synchronization is achieved, the data migration object can be safely deleted without host interruption.

**Important:** If performing an online migration, do not deactivate the data migration before deletion. Deactivating before deletion causes host I/O to stop, which can cause data corruption.
Right-click the data migration volume and select **Delete Data Migration** as shown in Figure 6-18. Deleting the migration can be done without host/server interruption.

**Figure 6-18  Delete Data Migration option**

**Restriction:** For safety purposes, you cannot delete an inactive or unsynchronized data migration from the Data Migration panel. An unfinished data migration can be deleted only by deleting the relevant volume from the Volumes → Volumes & Snapshots section in the XIV GUI.

### 6.4 Command-line interface

All of the XIV GUI operation steps can be performed using the XIV command-line interface (XCLI). You can use XCLI either through direct command execution or through batch files containing multiple commands. Using the CLI is especially helpful in migration scenarios involving numerous LUNs. This section lists the XCLI commands equivalent to the GUI steps shown previously. A full description of all the XCLI commands can be found in the [XCLI Users Guide](http://publib.boulder.ibm.com/infocenter/ibmxiv/r2/topic/com.ibm.help.xiv.doc/docs/GC27-2213-02.pdf) available at:


When using the XIV GUI, every command issued is logged in a text file with the correct syntax so you can review it later. This log is helpful for creating scripts. If you are running the XIV GUI under Microsoft Windows, look for a file named `guicommands_<todays date>.txt`. This file is typically found in the following folder:

C:\Documents and Settings\<Windows user ID >\Application Data\XIV\GUI\logs

The commands in the next few pages are in the order in which you must run them, starting with the commands to list all current definitions. You need these definitions when you delete migrations.

- **List targets.**
  
  **Syntax**  
  `target_list`

- **List target ports.**
  
  **Syntax**  
  `target_port_list`
List target connectivity.
Syntax
```plaintext
target_connectivity_list
```
Example
```
```
List clusters.
Syntax
```plaintext
cluster_list
```
List hosts.
Syntax
```plaintext
host_list
```
List volumes.
Syntax
```plaintext
vol_list
```
List data migrations.
Syntax
```plaintext
dm_list
```
Define target (Fibre Channel only).
Syntax
```plaintext
target_define target=<Name> protocol=FC xiv_features=no
```
Example
```
target_define target=DMX605 protocol=FC xiv_features=no
```
Define target port (Fibre Channel only).
Syntax
```plaintext
target_port_add fcaddress=<non-XIV storage WWPN> target=<Name>
```
Example
```
target_port_add fcaddress=0123456789012345 target=DMX605
```
Define target connectivity (Fibre Channel only).
Syntax
```plaintext
target_connectivity_define
local_port=1:FC_Port:<Module:Port> fcaddress=<non-XIV storage WWPN> target=<Name>
```
Example
```
target_connectivity_define local_port=1:FC_Port:5:4
fcaddress=0123456789012345 target=DMX605
```
Define cluster (optional).
Syntax
```plaintext
cluster_create cluster=<Name>
```
Example
```
cluster_create cluster=Exch01
```
Define host (if adding host to a cluster).
Syntax
```plaintext
host_define host=<Host Name> cluster=<Cluster Name>
```
Example
```
host_define host=Exch01N1 cluster=Exch01
```
Define host (if not using cluster definition).
Syntax
```plaintext
host_define host=<Name>
```
Example
```
host_define host=Exch01
```
Define host port (Fibre Channel host bus adapter port).
Syntax
```plaintext
host_add_port host=<Host Name> fcaddress=<HBA WWPN>
```
Example
```
host_add_port host=Exch01 fcaddress=123456789abcdef1
```
Create XIV volume using decimal GB volume size.
Syntax
```plaintext
vol_create vol=<Vol name> size=<Size> pool=<Pool Name>
```
Example
```
vol_create vol=Exch01_sg01_db size=17 pool=Exchange
```

Chapter 6. IBM XIV data migration 153
Create XIV volume using 512-byte blocks.

Syntax

```
vol_create vol=<Vol name> size_blocks=<Size in blocks>
   pool=<Pool Name>
```

Example

```
vol_create vol=Exch01_sg01_db size_blocks=32768
   pool=Exchange
```

Define data migration.

If you want the local volume to be automatically created:

Syntax

```
dm_define target=<Target> vol=<Volume Name>
   lun=<Host LUN ID as presented to XIV>
   source_updating=<yes|no>
   create_vol=yes pool=<XIV Pool Name>
```

Example

```
dm_define target=DMX605 vol=Exch01_sg01_db
   lun=5
   source_updating=no
   create_vol=yes pool=Exchange
```

If the local volume was pre-created:

Syntax

```
dm_define target=<Target> vol=<Pre-created Volume Name>
   lun=<Host LUN ID as presented to XIV>
   source_updating=<yes|no>
```

Example

```
dm_define target=DMX605 vol=Exch01_sg01_db
   lun=5
   source_updating=no
```

Test data migration object.

Syntax

```
dm_test vol=<DM Name>
```

Example

```
dm_test vol=Exch_sg01_db
```

Activate data migration object.

Syntax

```
dm_activate vol=<DM Name>
```

Example

```
dm_activate vol=Exch_sg01_db
```

Map volume to host/cluster.

- Map to host:

  Syntax

  ```
  map_vol host=<Host Name> vol=<Vol Name> lun=<LUN ID>
  ```

  Example

  ```
  map_vol host=Exch01 vol=Exch01_sg01_db
  ```

- Map to cluster:

  Syntax

  ```
  map_vol host=<Cluster Name> vol=<Vol Name> lun=<LUN ID>
  ```

  Example

  ```
  map_vol host=Exch01 vol=Exch01_sg01_db
  ```

Delete data migration object.

If the data migration is synchronized and thus completed:

Syntax

```
dm_delete vol=<DM Volume name>
```

Example

```
dm_delete vol=Exch01_sg01_db
```

If the data migration is not complete, delete it by removing the corresponding volume from the Volume and Snapshot menu or using the `vol_delete` command.

Delete volume (not normally needed).

This command must be acknowledged by the user before running. It therefore cannot be done using a script.

Syntax

```
vol_delete vol=<Vol Name>
```

Example

```
vol_delete vol=Exch_sg01_db
```
If you want to perform an unchallenged volume deletion:

Syntax

```
vol_delete -y vol=<Vol Name>
```

Example

```
vol_delete -y vol=Exch_sg01_db
```

- **Delete target connectivity.**

Syntax

```
target_connectivity_delete
local_port=1:FC_Port:<Module:Port> fcaddress=<non-XIV storage device WWPN> target=<Name>
```

Example

```
target_connectivity_delete local_port=1:FC_Port:5:4 fcaddress=0123456789012345 target=DMX605
```

- **Delete target port.**

Fibre Channel

Syntax

```
target_port_delete fcaddress=<non-XIV WWPN> target=<Name>
```

Example

```
target_port_delete fcaddress=0123456789012345 target=DMX605
```

- **Delete target.**

Syntax

```
target_delete target=<Target Name>
```

Example

```
target_delete target=DMX605
```

- **Change Migration Sync Rate**

Syntax

```
target_config_sync_rates target=<Target Name>
max_initialization_rate=<Rate in MB>
```

Example

```
target_config_sync_rates target=DMX605
max_initialization_rate=100
```

### 6.4.1 Using XCLI scripts or batch files

To run an XCLI batch job, use the XCLI rather than a background XCLI Session.

**Setting environment variables in Windows**

You can remove the need to specify user and password information for every command by making that information an environment variable at the Windows OS level. Example 6-1 shows how to set variables using a Windows command prompt. In the example, the XIV_XCLIUSER variable is set to `admin`, and the XIV_XCLIPASSWORD to `adminadmin`. Both variables are then confirmed as set. If necessary, change the user ID and password to suit your setup.

**Example 6-1 Setting environment variables in Microsoft Windows**

```
C:\>set XIV_XCLIUSER=admin
C:\>set XIV_XCLIPASSWORD=adminadmin
C:\>set | find "XIV"
XIV_XCLIPASSWORD=adminadmin
XIV_XCLIUSER=admin
```

To make these changes permanent, perform the following steps:

1. Right-click the My Computer icon and select **Properties**.
2. Click the **Advanced** tab.
3. Click **Environment Variables**.
4. Click **New** for a new system variable.
5. Create the XIV_XCLIUSER variable with the relevant user name.
6. Click **New** again to create the XIV_XCLIPASSWORD variable with the relevant password.

### Setting environment variables in UNIX

If you are using an operating system based on UNIX, export the environment variables as shown in Example 6-2, which in this example is AIX. The user and password variables are set to `admin` and `adminadmin`, and then confirmed as being set.

**Example 6-2   Setting environment variables in UNIX**

```
root@dolly:/tmp/XIVGUI# export XIV_XCLIUER=admin
root@dolly:/tmp/XIVGUI# export XIV_XCLIPASSWORD=adminadmin
root@dolly:/tmp/XIVGUI# env | grep XIV
XIV_XCLIPASSWORD=adminadmin
XIV_XCLIUER=admin
```

To make these changes permanent, update the relevant profile, making sure that you export the variables to make them environment variables.

**Tip:** It is also possible to run XCLI commands without setting environment variables with the `-u` and `-p` switches.

### 6.4.2 Sample scripts

With the environment variables set, a script or batch file like the one in Example 6-3 can be run from the shell or command prompt. The script defines the data migration pairings.

**Example 6-3   Data migration definition batch file**

```
xcli -m 10.10.0.10 dm_define vol=MigVol_1 target=DS4200_CTRL_A lun=4 source_updating=no create_vol=yes pool=test_pool
xcli -m 10.10.0.10 dm_define vol=MigVol_2 target=DS4200_CTRL_A lun=5 source_updating=no create_vol=yes pool=test_pool
xcli -m 10.10.0.10 dm_define vol=MigVol_3 target=DS4200_CTRL_A lun=7 source_updating=no create_vol=yes pool=test_pool
xcli -m 10.10.0.10 dm_define vol=MigVol_4 target=DS4200_CTRL_A lun=9 source_updating=no create_vol=yes pool=test_pool
xcli -m 10.10.0.10 dm_define vol=MigVol_5 target=DS4200_CTRL_A lun=11 source_updating=no create_vol=yes pool=test_pool
xcli -m 10.10.0.10 dm_define vol=MigVol_6 target=DS4200_CTRL_A lun=13 source_updating=no create_vol=yes pool=test_pool
xcli -m 10.10.0.10 dm_define vol=MigVol_7 target=DS4200_CTRL_A lun=15 source_updating=no create_vol=yes pool=test_pool
xcli -m 10.10.0.10 dm_define vol=MigVol_8 target=DS4200_CTRL_A lun=17 source_updating=no create_vol=yes pool=test_pool
xcli -m 10.10.0.10 dm_define vol=MigVol_9 target=DS4200_CTRL_A lun=19 source_updating=no create_vol=yes pool=test_pool
xcli -m 10.10.0.10 dm_define vol=MigVol_10 target=DS4200_CTRL_A lun=21 source_updating=no create_vol=yes pool=test_pool
```

Run an equivalent script or batch job to run the data migrations as shown in Example 6-4.

**Example 6-4   Activate data migration batch file**

```
xcli -m 10.10.0.10 dm_activate vol=MigVol_1
xcli -m 10.10.0.10 dm_activate vol=MigVol_2
```
6.5 Manually creating the migration volume

The local XIV volume can be created before defining the data migration object. However, this method is prone to manual calculation errors. These errors occur because it requires the size of the source volume on the non-XIV storage device in 512-byte blocks. The two volumes (source and XIV volume) must be exactly the same size. Finding the actual size of a volume in blocks or bytes can be difficult because certain storage devices do not show the exact volume size. You can rely on the host operating system to provide the real volume size, but this size is also not always reliable.

As an example, consider the ESS 800 volume 00F-FCA33 as depicted in Figure 6-26 on page 167. The size reported by the ESS 800 web GUI is 10 GB, which suggests that the volume is 10,000,000,000 bytes in size. The ESS 800 displays volume sizes using decimal counting. The AIX `bootinfo -s hdisk2` command reports the volume as 9,536 GiB, which is 9,999,220,736 bytes (1,073,741,824 bytes per GiB). Both of these values are too small. When the volume properties are viewed on the volume information window of the ESS 800 Copy Services GUI, it correctly reports the volume as being 19,531,264 sectors. This number is equivalent to 10,000,007,168 bytes (512 bytes per sector). When the XIV automatically creates a volume to migrate the contents of 00F-FCA33, it creates it as 19,531,264 blocks. Of the three information sources considered to calculate volume size, only one of them is correct. Using the automatic volume creation eliminates this uncertainty.

After you determine the exact size, select Blocks from the Volume Size list and enter the size of the XIV volume in blocks. If your sizing calculation is correct, you create an XIV volume that is the same size as the source (non-XIV storage device) volume. Then you can define a migration:

1. In the XIV GUI, click Remote → Migration.
2. Right-click Migration and select Define Data Migration. Make the appropriate entries and selections:
   - Destination Pool: Select the pool where the volume was created.
   - Destination Name: Select the pre-created volume from the list.
   - Source Target System: Select the already defined non-XIV storage device from the list.

   **Important:** If the non-XIV device is active/passive, the source target system must represent the controller (or service processor) on the device that owns the source LUN. You must find out from the non-XIV storage which controller is presenting the LUN to the XIV.
– **Source LUN**: Enter the decimal value of the LUN as presented to the XIV from the non-XIV storage system. Certain storage devices present the LUN ID as hex. The number in this field must be the decimal equivalent.

– **Keep Source Updated**: Select this check box if the non-XIV storage system source volume is to be updated with writes from the host. All writes from the host are written to the XIV volume and the non-XIV source volume until the data migration object is deleted.

3. Click **Define**.

4. Test the data migration object. Right-click the created data migration volume and select **Test Data Migration**. If there are any issues with the data migration object, the test fails and reports the issue found.

If the volume that you created is the wrong size, an error message is issued during the test data migration as shown in Figure 6-19. If you activate the migration, you get the same error message. You must delete the volume on the XIV and create a new, correctly sized one. It must be deleted because you cannot resize a volume that is in a data migration pair. In addition, you cannot delete a data migration pair unless it has completed the background copy. Delete the volume and then investigate why your size calculation was wrong. After correcting the problem, create a new volume and a new migration, and test it again.

![Figure 6-19 XIV volume wrong size for migration](image)

### 6.6 Changing and monitoring the progress of a migration

You can adjust the speed of the migration process to improve performance or reduce the impact of the migration on the XIV subsystem itself. You can also monitor its rate. Take care when planning the migration to avoid affecting the performance of the overall system and production I/O.
6.6.1 Changing the synchronization rate

There is only one tunable parameter that determines the speed at which migration data is transferred between the XIV and defined targets. There are two other tunable parameters that apply to XIV Remote Mirroring:

- **max_initialization_rate**
  
  This parameter sets the rate (in MBps) at which data is transferred between the XIV and the defined targets. The default rate is 100 MBps, and can be configured on a per-target basis. In this example, a total of 150 MBps (100+50) transfer rate is possible. If the transfer rate is lower than the initialization rate, you might be exceeding the capabilities of the non-XIV disk system. If the migration is not being done with attached hosts offline, consider dropping the initialization rate to a low number. Adjusting the rate ensures that the volume of migration I/O does not interfere with other hosts using the non-XIV disk system. You can then slowly increase the number while checking to ensure that response times are not affected on the other attached hosts. If you set the max_initialization_rate to zero, you stop the background copy, but hosts are still able to access all activated migration volumes.

- **max_syncjob_rate**
  
  This parameter (in MBps) is used in XIV remote mirroring for synchronizing mirrored snapshots. It is not normally relevant to data migrations. However, the max_initialization_rate cannot be greater than the max_syncjob_rate, which in turn cannot be greater than the max_resync_rate. In general, do not increase this rate.

- **max_resync_rate**
  
  This parameter (in MBps) is used for XIV remote mirroring only. It is not normally relevant to data migrations. This parameter defines the resync rate for mirrored pairs. After remotely mirrored volumes are synchronized, a resync is required if the replication is stopped for any reason. This parameter affects only this resync. The default rate is 300 MBps. There is no minimum or maximum rate. However, setting the value to 400 or more in a 4-Gbps environment does not show any increase in throughput. In general, do not increase this rate.

Increasing the max_initialization_rate parameter might decrease the time required to migrate the data. However, doing so might affect existing production servers on the non-XIV storage device. By increasing the rate parameters, more outgoing disk resources are used to serve migrations and less for existing production I/O. Be aware of how these parameters affect migrations and production. You can choose to use the higher rate only during off-peak production periods.

The rate parameters can only be set using XCLI, not the XIV GUI. Run the target_list -x command, where the -x parameter displays the current rate. If the setting is changed, the change takes place dynamically, so you do not need to deactivate or activate the migrations. As shown in Example 6-5, first display the target list and then confirm the current rates using the -x parameter. The example shows that the initialization rate is still set to the default value (100 MBps). You then increase the initialization rate to 200 MBps. You can then observe the completion rate, as shown in Figure 6-16 on page 151, to see whether it has improved.

**Example 6-5 Displaying and changing the maximum initialization rate**

```
>> target_list
Name                    SCSI Type   Connected
Nextrazap ITSO ESS800   FC          yes
>> target_list -x target="Nextrazap ITSO ESS800"
<XCLIRETURN STATUS="SUCCESS" COMMAND_LINE="target_list -x target=&quot;Nextrazap
ITSO ESS800&quot;">```

6.6.2 Monitoring migration speed

If you want to monitor the speed of the migration you can use the Data Migration window as shown in Figure 6-16 on page 151. The status bar can be toggled between GB remaining, percent complete, or hours/minutes remaining. However, if you want to monitor the actual MBps, you must use an external tool. An external tool is needed is because the performance statistics displayed using the XIV GUI or using an XIV tool do not include data migration I/O. They do, however, show incoming I/O rates from hosts using LUNs that are being migrated.

6.6.3 Monitoring migration using the XIV event log

The XIV event log can be used to confirm when a migration started and finished using the following steps:

1. From the XIV GUI, click Monitor → Events.
2. On the Events window, select dm in the Type list and click Filter.
Figure 6-20 shows the events for a single migration. In this example, the events must be read from bottom to top. You can sort the events by date and time by clicking the **Date** column in the Events window.

### 6.6.4 Monitoring migration speed through the fabric

If you have a Brocade-based SAN, use the `portperfshow` command and verify the throughput rate of the initiator ports on the XIV. If you have two fabrics, you might need to connect to two different switches. If multiple paths are defined between XIV and non-XIV disk system, the XIV load balances across those ports. This balancing means that you must aggregate the throughput numbers from each initiator port to see total throughput. Example 6-6 shows the output of the `portperfshow` command. The values shown are the combined send and receive throughput in MBps for each port. In this example, port 0 is the XIV Initiator port and port 1 is a DS4800 host port. The `max_initialization_rate` is set to 50 MBps.

**Example 6-6** Brocade `portperfshow` command

```
FB1_RC6_PDC:admin> portperfshow
0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15  Total
======================================================================================
50m 50m 14m 14m 2.4m 848k 108k 34k  0  937k 0 27m 3.0m  0  949k 3.0m 125m
```

If you have a Cisco-based SAN, start Device Manager for the relevant switch and then select **Interface → Monitor → FC Enabled**.

### 6.6.5 Monitoring migration speed through the non-XIV storage

The ability to display migration throughput varies by non-XIV storage device. For example, if you are migrating from a DS4000, you can use the performance monitoring windows in the DS4000 System Manager. In the DS4000 System Manager GUI, click **Storage Subsystem → Monitor Performance** to show the volumes being migrated and the throughput for the relevant controllers. You can then determine what percentage of I/O is
being generated by the migration process. In Figure 6-21, you can see that one volume is being migrated using a max_initialization_rate of 50 MBps. This volume represents the bulk of the I/O being serviced by the DS4000 in this example.

![Figure 6-21 Monitoring a DS4000 migration](image)

## 6.7 Thick-to-thin migration

When the XIV migrates data from a LUN on a non-XIV disk system to an XIV volume, it reads every block of the source LUN. However, when it comes to writing this data into the XIV volume, the XIV writes only blocks that contain data. Blocks that contain only zeros are not written and do not take any space on the XIV. This process is called a thick-to-thin migration, and occurs regardless of whether you are migrating the data into a thin provisioning pool or a regular pool.

During the migration, the value displayed in the Used column of the Volumes and Snapshots window drops every time empty blocks are detected. When the migration is completed, you can check this column to determine how much real data was written into the XIV volume. In Figure 6-22, the used space on the Windows2003_D volume is 4 GB. However, the Windows file system using this disk shown in Figure 6-24 on page 164 shows only 1.4 GB of data.

![Figure 6-22 Thick-to-thin results](image)

This discrepancy occurs because when file deletions occur at a file system level, the data is not removed. The file system reuses this effectively free space. However, the system does not write zeros over the old data because doing so generates a large amount of unnecessary I/O. The result is that the XIV copies old and deleted data during the migration. Copying this obsolete data makes no difference to the migration speed because the blocks must be read into the cache regardless of what they contain.

If you are not planning to use the thin provisioning capability of the XIV, this is not an issue.

### 6.7.1 Writing zeros to recover space

One way to recover space before you start a migration is to use a utility to write zeros across all free space. In a UNIX environment, use a simple script like the one shown in Example 6-7 to write large empty files across your file system. You might need to run these commands many times to use up all the empty space.

```
Example 6-7 Writing zeros across your file system

# The next command will write a 1 GB mytestfile.out
dd if=/dev/zero of=mytestfile.out bs=1000 count=1000000
```
# The next command will free the file allocation space
rm mytestfile.out

In a Windows environment, you can use a Microsoft tool known as *sdelete* to write zeros across deleted files. You can find this tool in the sysinternals section of Microsoft Technet at the following web address:


6.7.2 Recovering space after the migration

If you instead choose to write zeros to recover space after the migration, you must initially generate large amounts of empty files. It takes several days for the used space value to decrease after the script or application is run. It takes extra time for because recovery of empty space runs as a background task.

6.8 Resizing the XIV volume after migration

Because of the way that XIV distributes data, the XIV allocates space in 17-GB portions (which are exactly 17,179,869,184 bytes or 16 GiB). When creating volumes using the XIV GUI, any volume size you enter gets rounded up to the next 17-GB cutoff.

If you to allow the XIV to determine the size of the migration volume, a small amount of extra space is consumed for every volume that was created. The volumes automatically created by the XIV often reserve more XIV disk space than is made available to the volume. Avoid this problem by creating volume sizes on the non-XIV storage device in multiples of 16 GiB. An example of the XIV volume properties of such an automatically created volume is shown in Figure 6-23. In this example, the Windows2003_D drive is 53 GB in size, but the size on disk is 68 GB on the XIV.

![Figure 6-23 - Properties of a migrated volume](image)
This volume can be resized to 68 GB, which makes the volume 15 GB larger without consuming any more space on the XIV. Figure 6-24 shows that the migrated Windows2003_D drive is 53 GB in size (53,678,141,440 bytes).

Figure 6-24  Windows D drive at 53 GB

To resize a volume, perform the following steps:

1. Click Volumes → Volumes & Snapshots.
2. Right-click to select the volume and select Resize.
3. Change the sizing method from Blocks to GB.

   The volume size is automatically moved to the next multiple of 17 GB.

You can also use XCLI commands as shown in Example 6-8.

Example 6-8   Resize the D drive using XCLI

```
>> vol_resize vol=Windows2003_D size=68
```

Warning:  ARE_YOU_SURE_YOU_WANT_TO_ENLARGE_VOLUME Y/N: Y
Command executed successfully.

Because this example is for a Microsoft Windows 2003 basic NTFS disk, you can use the diskpart utility to extend the volume (Example 6-9).

Example 6-9   Expanding a Windows volume

```
C:\>diskpart
DISKPART> list volume

<table>
<thead>
<tr>
<th>Volume ###</th>
<th>Ltr</th>
<th>Label</th>
<th>Fs</th>
<th>Type</th>
<th>Size</th>
<th>Status</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volume 0</td>
<td>C</td>
<td></td>
<td>NTFS</td>
<td>Partition</td>
<td>34 GB</td>
<td>Healthy</td>
<td>System</td>
</tr>
<tr>
<td>Volume 4</td>
<td>D</td>
<td>Windows2003</td>
<td>NTFS</td>
<td>Partition</td>
<td>64 GB</td>
<td>Healthy</td>
<td></td>
</tr>
</tbody>
</table>

DISKPART> select volume 4
```
Volume 4 is the selected volume.

DISKPART> **extend**

DiskPart successfully extended the volume

Confirm that the volume has grown by displaying the volume properties. In Figure 6-25, you can see that the disk is now 68 GB (68,713,955,328 bytes).

![Figure 6-25 Windows 2003 D drive has grown to 64 GB](image)

In terms of **when** to do the resize, a volume cannot be resized while it is part of a data migration. The migration process must complete and the migration for that volume must be deleted before the volume can be resized. For this reason, you might choose to defer resizing until after the migration of all relevant volumes is complete. This technique also separates the resize change from the migration change. Depending on the operating system using that volume, you might not get any benefit from doing this resize.

### 6.9 Troubleshooting

This section lists common errors that are encountered during data migrations using the XIV data migration facility.

#### 6.9.1 Target connectivity fails

The connections (link line) between the XIV and non-XIV disks system on the migration connectivity panel remain colored red or the link shows as **down**. There are several reasons that the connection can fail:

- On the Migration Connectivity window, verify that the status of the XIV initiator port is **OK (Online)**. If not, check the connections between the XIV and the SAN switch.
- Verify that the Fibre Channel ports on the non-XIV storage device are set to target, enabled, and online.
- Check whether SAN zoning is incorrect or incomplete. Verify that SAN fabric zoning configuration for XIV and non-XIV storage device are active.
Check the SAN switch name server that both XIV ports and non-XIV storage ports are logged in correctly. Verify that XIV and non-XIV are logged in to the switch at the correct speed.

Check whether the XIV WWPN is not properly defined to the non-XIV storage device target port. The XIV WWPN must be defined as a Linux or Windows host:

- If the XIV initiator port is defined as a Linux host to the non-XIV storage device, change the definition to a Windows host. Delete the link (line connections) between the XIV and non-XIV storage device ports and redefine the link. This step is storage device dependent. The difference is caused by how the non-XIV storage device presents a pseudo LUN-0 if a real volume is not presented as LUN 0.

- If the XIV initiator port is defined as a Windows host to the non-XIV storage device, change the definition to a Linux host. Delete the link (line connections) between the XIV and non-XIV storage device ports and redefine the link. This step is storage device dependent. The difference is caused by how the non-XIV storage device presents a pseudo LUN-0 if a real volume is not presented as LUN 0.

- If the previous two attempts are not successful, assign a real disk/volume to LUN 0 and present to the XIV. The volume assigned to LUN-0 can be a small unused volume or a real volume that will be migrated.

Offline/Online the XIV Fibre Channel port:

a. Go to the Migration Connectivity window.
b. Click the link between XIV and the target system.
c. Right-click the port in question and select Configure.
d. Select No in the Enabled list.
e. Click Configure.
f. Repeat the process, selecting Yes for Enabled.

Change the port type from Initiator to Target and then back to Initiator. This forces the port to completely reset and reload.

a. Go to the Migration Connectivity window.
b. Clicking the link between the XIV and target system.
c. Right-click the port in question and select Configure.
d. Select Target in the Role list.
e. Click Configure.
f. Repeat the process, selecting Initiator for the role.

6.9.2 Remote volume LUN is unavailable

This error typically occurs when defining a data migration volume and the LUN ID specified in the Source LUN field is not responding to the XIV. The error can occur for several reasons:

- The LUN ID (host LUN ID or SCSI ID) is not allocated to the XIV on the ports identified in the target definition. Log on to the non-XIV storage device to confirm.

- The LUN ID is not allocated to the XIV on all ports specified in the target definition. For example, if the target definition has two links, the volume must be allocated down both paths using the same LUN ID. The XIV looks for the LUN ID specified on the first defined path. If it does not have access to the LUN, it fails even if the LUN is allocated down the second path. If two links are defined from the target (non-XIV) storage device to the XIV, the LUN must be allocated down both paths.

- The LUN ID is incorrect. Do not confuse the internal LUN ID of a non-XIV storage device with the SCSI LUN ID (host LUN ID) that is presented to the XIV. The source LUN must be the LUN ID (decimal) as presented to the XIV.
The Source LUN ID field is expecting a decimal number. Certain vendors present the LUN ID in hex, must be translated to decimal. Therefore, if LUN ID 10 displays its IDs in hex, the LUN ID in the data migration define is 16 (hex 10). An example of a hexadecimal LUN number is shown in Figure 6-26 on page 167. In this example you can see LUN 000E, 000F, and 0010 on an ESS 800. These LUNs are entered into the XIV data migration definitions as LUNs 14, 15, and 16. See 6.12, “Other considerations” on page 173, for more details.

The LUN ID allocated to the XIV is allocated to an incorrect XIV WWPN. Make sure that the volume is allocated to the correct XIV WWPNs.

If multiple data migration targets are defined, the wrong target might have been chosen when the data migration was defined.

Sometimes when volumes are added after the initial connectivity is defined, the volume is not available. Go to the Migration Connectivity window and delete the links between the XIV and non-XIV storage device and recreate them. Go back to the DM window and recreate the data migration. For more information, see “Define non-XIV storage device to XIV (as a migration target)” on page 141.

The volume on the source non-XIV storage device might not be initialized or been low-level formatted. If the volume has data on it, this is not the case. However, if you are assigning new volumes from the non-XIV storage device, these new volumes might not complete the initialization process. On ESS 800 storage, the initialization process can be displayed from the Modify Volume Assignments window. Figure 6-26 shows the volumes are still 0% background formatted, so they are not accessible by the XIV. For ESS 800, keep clicking Refresh Status on the ESS 800 web GUI until the formatting message disappears.

### 6.9.3 Local volume is not formatted

This error occurs when an existing volume is chosen as the destination name that is already written either from a host or a previous DM process that was removed. To get around this error, perform one of the following actions:

- Use another volume as a migration destination.
- Delete the volume that you are trying to migrate to and then create it again.
- Click Volumes → Volumes and Snapshots, right-click the volume and select Format.
6.9.4 Host server cannot access the XIV migration volume

This error occurs if you read the contents of a volume on a non-XIV storage device using an XIV data migration without activating the data migration. It also happens if the migration is performed without following the correct order of steps. Do not attempt to access the XIV volume being migrated until the XIV shows that the migration is initializing and active, or fully synchronized.

**Tip:** This error might also happen in a cluster environment where the XIV is holding a SCSI reservation. Make sure that all nodes of a cluster are shut down before starting a migration. The XCLI command `reservation_list` lists all SCSI reservations held by the XIV. If you find a volume with reservations where all nodes are offline, remove the reservations using the XCLI command `reservation_clear`. See the XCLI documentation for further details.

6.9.5 Remote volume cannot be read

This error occurs when a volume is defined down the passive path on an active/passive multi-pathing storage device. It can occur in several cases:

- Two paths were defined on a target (non-XIV storage device) that supports only active/passive multi-pathing. XIV is an active/active storage device. Defining two paths on any target from an active/passive multi-pathing storage device is not supported. Define the target with only one path. Another target can be defined with one connection to the other controller. For example, consider an environment where the non-XIV storage device has two controllers, but the volume can be active on only one at time. In this environment, controller A can be defined as one target on the XIV and controller B can be defined as a separate target. In this manner, all active volumes on controller A can be migrated down the XIV A target. Similarly, all volumes active on the B controller can be migrated down the XIV B target.

- When defining the XIV initiator to an active/passive multi-pathing non-XIV storage device, certain storage devices allow the initiator to be defined as *not* supporting failover. Configure the XIV initiator to the non-XIV storage device in this manner. When configured as such, the volume on the passive controller is not presented to the initiator (XIV). The volume is only presented down the active controller.

For more information, see 6.2.2, “Multi-pathing with data migrations” on page 137, and 6.12, “Other considerations” on page 173.

6.9.6 LUN is out of range

XIV currently supports migrating data from LUNs with a LUN ID less than 513 (decimal). This limitation is not usually an issue because most non-XIV storage devices present volumes on an initiator basis. If three hosts are connected to the same port on a non-XIV storage device, each host can be allocated volumes starting at the same LUN ID. For migration purposes, you must do one of the following procedures:

- Map one host at a time (and then reuse the LUN IDs for the next host)
- Use different sequential LUN numbers
If three hosts each have three LUNs mapped using LUN IDs 20, 21, and 22, for migration purposes, migrate them using the following LUN IDs:

- IDs 30, 31, 32 (first host)
- IDs 33, 34, 35 (second host)
- IDs 36, 37, 38 (third host)

From the XIV you can again map them to each host as LUN IDs 20, 21, and 22 (as they were from the non-XIV storage).

If migrating from an EMC Symmetrix or DMX, there are special considerations. See 6.12, “Other considerations” on page 173.

6.10 Backing out of a data migration

For change management purposes, you might be required to document a back-out procedure. There are four possible points in the migration process where a back-out might occur:

- Back out before migration is defined on the XIV
- Back out after a data migration is defined but not activated
- Back out after a data migration is activated but is not complete
- Back out after a data migration reaches the synchronized state

6.10.1 Back out before migration is defined on the XIV

If a data migration definition does not exist yet, then no action must be taken on the XIV. You can zone the host server back to the non-XIV storage system and map LUNs on the host server back to the host server. Make sure that the correct LUN order is preserved.

6.10.2 Back out after a data migration is defined but not activated

If the data migration definition exists but is not activated, follow the steps described in 6.10.1, “Back out before migration is defined on the XIV” on page 169. To remove the inactive migration from the migration list, delete the XIV volume that was going to receive the migrated data.

6.10.3 Back out after a data migration is activated but is not complete

If the data migration status is initialization in the GUI or the XCLI shows it as active=yes, the background copy process is started. Deactivating the migration in this state blocks any I/O passing through the XIV from the host server to the LUNs on the XIV and non-XIV systems. To back out, first shut down the host server or its applications. Then deactivate the data migration and delete the XIV data migration volume if wanted. Finally, restore the original LUN masking and SAN fabric zoning and bring your host back up.

Important: Choosing to not allow source updating and write I/O occurs after the migration started prevents the LUN on the non-XIV storage device from containing those writes.
6.10.4 Back out after a data migration reaches the synchronized state

If the data migration status in the GUI is synchronized, the background copy is complete. In this case backout can still occur because the data migration is not destructive to the source LUN on the non-XIV storage device. Reverse the process by shutting down the host server or applications and restoring the original LUN masking and switch zoning settings. You might need to also reinstall the relevant host server multi-path software for access to the non-XIV storage device.

**Important:** If you chose to not allow source updating and write I/O has occurred during or after the migration, the LUN on the non-XIV storage device will not contain those writes.

6.11 Migration checklist

There are three separate stages to a migration:

1. Prepare the environment for the implementation of the XIV
2. Cut over your hosts
3. Remove any old devices and definitions

For site setup, the high-level process includes the following steps:

1. Install XIV and cable it into the SAN.
2. Pre-populate SAN zones in switches.
3. Pre-populate the host/cluster definitions in the XIV.
4. Define XIV to the non-XIV disk as a host.
5. Define the non-XIV disk to XIV as a migration target.
6. Confirm paths.

For each host, the high-level process includes the following steps:

1. Update host drivers, install Host Attachment Kit, and shut down the host.
2. Disconnect/Un-Zone the host from non-XIV storage.
3. Zone the host to XIV.
4. Map the host LUNs away from the host instead of mapping them to the XIV.
5. Create XIV data migration.
6. Map XIV data migration volumes to the host.
7. Start the host.

When all data on the non-XIV disk system is migrated, perform site cleanup using these steps:

1. Delete all SAN zones related to the non-XIV disk.
2. Delete all LUNs on non-XIV disk and remove it from the site.
Table 6-1 shows the site setup checklist.

<table>
<thead>
<tr>
<th>Task Number</th>
<th>Completed</th>
<th>Where to perform</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>Site</td>
<td>Install XIV.</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>Site</td>
<td>Run fiber cables from SAN switches to XIV for host connections and migration connections.</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>Non-XIV storage</td>
<td>Select host ports on the non-XIV storage to be used for migration traffic. These ports do not have to be dedicated ports. Run new cables if necessary.</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>Fabric switches</td>
<td>Create switch aliases for each XIV Fibre Channel port and any new non-XIV ports added to the fabric.</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>Fabric switches</td>
<td>Define SAN zones to connect hosts to XIV, but do not activate the zones. Define them by cloning the existing zones from host to non-XIV disk and swapping non-XIV aliases for new XIV aliases.</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>Fabric switches</td>
<td>Define and activate SAN zones to connect non-XIV storage to XIV initiator ports (unless direct connected).</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>Non-XIV storage</td>
<td>If necessary, create a small LUN to be used as LUN0 to allocate to the XIV.</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>Non-XIV storage</td>
<td>Define the XIV on the non-XIV storage device, mapping LUN0 to test the link.</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>XIV</td>
<td>Define non-XIV storage to the XIV as a migration target and add ports. Confirm that links are green and working.</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>XIV</td>
<td>Change the max_initialization_rate depending on the non-XIV disk. You might want to start at a smaller value and increase it if no issues are seen.</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>XIV</td>
<td>Define all the host servers to the XIV (cluster first if using clustered hosts). Use a host listing from the non-XIV disk to get the WWPNs for each host.</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>XIV</td>
<td>Create storage pools as required. Ensure that there is enough pool space for all the non-XIV disk LUNs being migrated.</td>
</tr>
</tbody>
</table>

After the site setup is complete, the host migrations can begin. Table 6-2 shows the host migration checklist. Repeat this checklist for every host. Task numbers identified with an asterisk must be performed with the host application offline.

<table>
<thead>
<tr>
<th>Task number</th>
<th>Completed?</th>
<th>Where to perform</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>Host</td>
<td>From the host, determine the volumes to be migrated and their relevant LUN IDs and hardware serial numbers or identifiers.</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>Host</td>
<td>If the host is remote from your location, confirm that you can power the host back on after shutting it down. You can use tools such as an RSA card or IBM BladeCenter® manager.</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>Non-XIV Storage</td>
<td>Get the LUN IDs of the LUNs to be migrated from non-XIV storage device. Convert from hex to decimal if necessary.</td>
</tr>
<tr>
<td>Task number</td>
<td>Completed?</td>
<td>Where to perform</td>
<td>Task</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------</td>
<td>------------------</td>
<td>------</td>
</tr>
<tr>
<td>4*</td>
<td>Host</td>
<td>Shut down the application.</td>
<td></td>
</tr>
<tr>
<td>5*</td>
<td>Host</td>
<td>Set the application to <em>not</em> start automatically at reboot. Using this setting helps when performing administrative functions on the server such as upgrades of drivers, patches, and so on.</td>
<td></td>
</tr>
<tr>
<td>6*</td>
<td>Host</td>
<td>On UNIX servers, comment out disk mount points on affected disks in the mount configuration file. Commenting them out helps with system reboots while configuring for XIV.</td>
<td></td>
</tr>
<tr>
<td>7*</td>
<td>Host</td>
<td>Shut down affected servers.</td>
<td></td>
</tr>
<tr>
<td>8*</td>
<td>Fabric</td>
<td>Change the active zone set to exclude the SAN zone that connects the host server to non-XIV storage. In addition, include the SAN zone for the host server to XIV storage. Create the new zone during site setup.</td>
<td></td>
</tr>
<tr>
<td>9*</td>
<td>Non-XIV storage</td>
<td>Unmap source volumes from the host server.</td>
<td></td>
</tr>
<tr>
<td>10*</td>
<td>Non-XIV storage</td>
<td>Map source volumes to the XIV host definition created during site setup.</td>
<td></td>
</tr>
<tr>
<td>11*</td>
<td>XIV</td>
<td>Create data migration pairing (XIV volumes created dynamically).</td>
<td></td>
</tr>
<tr>
<td>12*</td>
<td>XIV</td>
<td>Test XIV migration for each volume.</td>
<td></td>
</tr>
<tr>
<td>13*</td>
<td>XIV</td>
<td>Start XIV migration and verify it. If you want, wait for migration to finish.</td>
<td></td>
</tr>
<tr>
<td>14*</td>
<td>Host</td>
<td>Boot the server. Be sure that the server is <em>not</em> attached to any storage.</td>
<td></td>
</tr>
<tr>
<td>15*</td>
<td>Host</td>
<td>Coexistence of non-XIV and XIV multi-pathing software is supported with an approved SCORE(RPQ) only. Remove any unapproved multi-pathing software.</td>
<td></td>
</tr>
<tr>
<td>16*</td>
<td>Host</td>
<td>Install patches, update drivers, and HBA firmware as necessary.</td>
<td></td>
</tr>
<tr>
<td>17*</td>
<td>Host</td>
<td>Install the XIV Host Attachment Kit. Be sure to note the prerequisites.</td>
<td></td>
</tr>
<tr>
<td>18*</td>
<td>Host</td>
<td>You might need to reboot depending on the operating system.</td>
<td></td>
</tr>
<tr>
<td>19*</td>
<td>XIV</td>
<td>Map XIV volumes to the host server. (Use original LUN IDs.)</td>
<td></td>
</tr>
<tr>
<td>20*</td>
<td>Host</td>
<td>Reboot server</td>
<td></td>
</tr>
<tr>
<td>21*</td>
<td>Host</td>
<td>Verify that the LUNs are available and that pathing is correct.</td>
<td></td>
</tr>
<tr>
<td>22*</td>
<td>Host</td>
<td>For UNIX servers, update the mount points for new disks in the mount configuration file if they have changed. Mount the file systems.</td>
<td></td>
</tr>
<tr>
<td>23*</td>
<td>Host</td>
<td>Start the application.</td>
<td></td>
</tr>
<tr>
<td>24*</td>
<td>Host</td>
<td>Set the application to start automatically if this setting was previously changed.</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>XIV</td>
<td>Monitor the migration if it is not already completed.</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>XIV</td>
<td>When the volume is synchronized, delete the data migration. Do <em>not</em> deactivate the migration.</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>Non-XIV Storage</td>
<td>Unmap migration volumes away from XIV if you need to free up LUN IDs.</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>XIV</td>
<td>Consider resizing the migrated volumes to the next 17 GB boundary if the host operating system is able to use new space on a resized volume.</td>
<td></td>
</tr>
</tbody>
</table>
When all the hosts and volumes are migrated, perform the site cleanup tasks shown in Table 6-3.

<table>
<thead>
<tr>
<th>Task number</th>
<th>Completed?</th>
<th>Where to perform</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>29</td>
<td></td>
<td>Host</td>
<td>If XIV volume was resized, use host procedures to use the extra space.</td>
</tr>
<tr>
<td>30</td>
<td></td>
<td>Host</td>
<td>If non-XIV storage device drivers and other supporting software were not removed earlier, remove them when convenient.</td>
</tr>
</tbody>
</table>

When all the hosts and volumes are migrated, perform the site cleanup tasks shown in Table 6-3.

**Table 6-3  Site cleanup checklist**

<table>
<thead>
<tr>
<th>Task number</th>
<th>Completed?</th>
<th>Where to perform</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>XIV</td>
<td>Delete migration paths and targets.</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>Fabric</td>
<td>Delete all zones related to non-XIV storage, including the zone for XIV migration.</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>Non-XIV storage</td>
<td>Delete all LUNs and perform secure data destruction if required.</td>
</tr>
</tbody>
</table>

**6.12 Other considerations**

The XIV supports migration from practically any SCSI storage device that has Fibre Channel interfaces. To ensure a successful migration from your specific storage device to XIV, the following items must be validated:

- LUN0: Do you need to specifically map a LUN to LUN ID zero? This determines whether you will have a problem defining the paths.
- LUN numbering: Does the storage device GUI or CLI use decimal or hexadecimal LUN numbering? This determines whether you must do a conversion when entering LUN numbers into the XIV GUI.
- Multipathing: Is the device active/active or active/passive? This determines whether you define the storage device as a single target, or as one target per internal controller or service processor.
- Definitions: Does the device have specific requirements when defining hosts?

For more information, see *IBM XIV Storage System: Copy Services and Migration*, SG24-7759, at:

This chapter addresses various methods for migrating data from any SAN Volume Controller supported storage subsystem to a different one. The storage types covered include between single or multiple, and similar or dissimilar.

This chapter provides a brief description of the IBM System Storage SAN Volume Controller, SAN Volume Controller terminology, and architecture.

In October 2010, IBM introduced a new midrange disk system called IBM Storwize® V7000. V7000 is based on SAN Volume Controller virtualization technology and provides the same functions and interoperability as SAN Volume Controller.

For more information about Storwize, see the following web address:
http://www-03.ibm.com/systems/storage/disk/storwize_v7000/

This chapter contains the following sections:

- IBM System Storage SAN Volume Controller overview
- SAN Volume Controller concepts for migrating the data
- Migrating using SAN Volume Controller
- SAN Volume Controller Migration preparation prerequisites
- Migrating SAN disks to SAN Volume Controller volumes and back to SAN
- SAN Volume Controller Volume migration between two storage pools
- Data migration using SAN Volume Controller mirrored volumes
- Data migration using SAN Volume Controller Metro Mirror
- SAN Volume Controller as data migration engine
- Other resources
7.1 IBM System Storage SAN Volume Controller overview

The IBM System Storage SAN Volume Controller is a scalable hardware and software solution. It provides block aggregation and logical drive management for different disk storage subsystems in a SAN environment. SAN Volume Controller provides the following advantages:

- A single view of the storage attached to the SAN: You can manage, add, and migrate physical disks non-disruptively, even between storage subsystems.

**Tip:** The SAN must be zoned in such a way that the application servers cannot see the storage. This zoning prevents conflict between the SAN Volume Controller and the application servers that are both trying to manage the storage.

- Storage virtualization: SAN Volume Controller creates a storage pool of managed disks from attached disk storage subsystems. These managed disks are then mapped to a set of volumes for use by host computer systems.

- Scalable: SAN Volume Controller can be used to manage all of your disk storage requirements, or just a subset of them. SAN Volume Controller also offers a large scalable cache using an algorithm.

- Reduces the requirement for additional partitions: SAN Volume Controller consumes only one storage partition for each storage server that connects to it.

- Improves access: SAN Volume Controller improves capacity utilization, and spare capacity. Underlying physical disks can be reallocated non-disruptively from an application server point of view irrespective of the server operating system or platform type.

- Simplifies device driver configuration on hosts: All hosts within your network use the same IBM device driver to access all storage subsystems through the SAN Volume Controller.

- Supports split I/O group implementations: Split implementations are not apparent to an application and are used across sites to cover application high availability requirements. These implementations can be used in case of high availability demands.

SAN Volume Controller is licensed according to the usable capacity that is being managed. The advanced functions available on SAN Volume Controller, such as FlashCopy, IBM Easy Tier®, Split I/O group, Mirrored volumes, Metro Mirror, and Global Mirror are included.

The license cost is for the capacity of all storage managed by the SAN Volume Controller, plus the capacity of the copy services maintained by the SAN Volume Controller. You can upgrade at any time by purchasing a license for the additional capacity required.

SAN Volume Controller supports a wide variety of disk storage and host operating system platforms. For the latest information, see the following web address:
http://www-03.ibm.com/systems/storage/software/virtualization/svc/interop.html

For details and information about SAN Volume Controller implementation, see *Implementing the IBM System Storage SAN Volume Controller V6.1*, SG24-7933.
7.1.1 SAN Volume Controller components, concepts, and terminology

An SAN Volume Controller implementation consists of both hardware and software. The hardware consists of the following items:

- A management console
- A minimum of two node pairs to form the SAN Volume Controller cluster
- A minimum of two uninterruptible power supplies (UPSs)

According to new hardware and software releases (at the writing of this book, release 6.2), SAN Volume Controller supports the following additional new functions:

- Support for 10 Gbps iSCSI functionality with 2145-CG8 node hardware
- Real-time performance monitoring
- Support for VMware vStorage APIs for Array Integration (VAAI)
- Internal SSD drive support
- Support for FlashCopy targets as Metro or Global Mirror sources
- Critical update notifications
- Additional language support
- New management GUI
- No requirement for separate console installation
- Direct cluster access using a web browser
- Easy Tier hotspot management
- Service Assistant

**Remember:** Depending on your SAN Volume Controller code level, some functions might not be available.

SAN Volume Controller naming terminology has recently changed as shown in Table 7-1.

<table>
<thead>
<tr>
<th>SAN Volume Controller 5.1</th>
<th>SAN Volume Controller 6.1 and later</th>
</tr>
</thead>
<tbody>
<tr>
<td>Managed Disk Group</td>
<td>Storage pool</td>
</tr>
<tr>
<td>Virtual Disk (VDisk)</td>
<td>Volume</td>
</tr>
<tr>
<td>Space-efficient</td>
<td>Thin provisioning</td>
</tr>
<tr>
<td>VDisk-to-host mapping</td>
<td>Host mapping</td>
</tr>
<tr>
<td>Error</td>
<td>Event</td>
</tr>
</tbody>
</table>
SAN Volume Controller management
With the new SAN Volume Controller code release, you can access SAN Volume Controller nodes directly through a web browser or using the command-line interface (CLI). It is no longer necessary to have an SAN Volume Controller Console installed. The web GUI is embedded in the SAN Volume Controller code and can be accessed directly from any workstation using a web browser. The IBM SSPC (System Storage Productivity Center) server is still in place and is valuable as central point of management.

Node
A node is an individual server in a SAN Volume Controller cluster on which the SAN Volume Controller software runs. Nodes are always installed as pairs and represent one I/O group in the SAN Volume Controller cluster concept. Each node is connected to its own UPS to ensure a safe power off during a power outage.

Configuration node
At any time, a single node in the cluster is used to manage configuration activity. This configuration node manages an information cache that describes the cluster configuration and provides a focal point for configuration commands.

Similarly, at any time, a single node is responsible for the overall management of the cluster. In a configuration node failure, the SAN Volume Controller fails over to the second node, which takes the configuration function. The IP address and user credentials remain the same.

SAN Volume Controller cluster
When the first two nodes are installed, they form an SAN Volume Controller cluster that can contain up to eight nodes (four pairs of nodes).

I/O group
An input/output (I/O) group contains two SAN Volume Controller nodes defined by the configuration process. Each SAN Volume Controller node is associated with exactly one I/O group. The nodes in the I/O group provide access to the volumes in the I/O group.

Internal storage and external storage
SAN Volume Controller system can manage a combination of internal and external storage:

- Internal storage: SAN Volume Controller model 2145-CF8 and model 2145-CG8 support up to eight SSD drives attached to them per I/O group. These drives are used to create a Redundant Array of Independent Disks (RAID) that is presented as managed disks (MDisks) to the system.
- External storage: The SAN Volume Controller can detect logical units (LUs) on an external storage system that is attached through Fibre Channel connections. These LUs are detected as managed disks (MDisks) in the system, and must be protected from drive failures using RAID technology on the storage system.

Front-End and back-end
SAN Volume Controller takes managed disks from back-end storage subsystem, groups them in storage pools, virtualizes them, and presents them to application servers (hosts) as volumes.

The switch fabric must be zoned, so that the SAN Volume Controller nodes can detect the back-end storage systems and the front-end host HBAs. The SAN Volume Controller prevents direct host access to managed disks. The managed disks are looked after by the back-end
application of the SAN Volume Controller. The volumes presented to hosts are looked after by the front-end application in the SAN Volume Controller.

**Managed disks (MDisks)**
A managed disk (MDisk) is a disk presented by a storage subsystem and managed by the SAN Volume Controller. This MDisk must be on a RAID storage subsystem. An MDisk provides usable blocks (or extents) of physical storage to the SAN Volume Controller cluster. The MDisks must be zoned in a way that they are not visible to host systems on the SAN. An MDisk in the SAN Volume Controller can be either managed or unmanaged. A managed MDisk is an MDisk assigned to a storage pool.

**Storage pool**
A storage pool is a collection of MDisks that jointly contain all the data for a specified set of volumes. Each storage pool is divided into a number of extents. When creating a storage pool, you must choose an extent size. After it is set, the extent size stays constant for the life of that storage pool. Each storage pool can have a different extent size.

**Volumes**
A volume is a logical entity that represents extents contained in one or more MDisks from a storage pool. Volumes are allocated in whole numbers of extents. Each volume is only associated with a single I/O group. The volume is then presented to the host as a LUN for use.

There are three types of volumes:
- Striped: A volume created in striped mode has extents allocated from each MDisk in the storage pool in a round-robin fashion.
- Sequential: The volumes are allocated sequentially on one MDisk to create the volume.
- Image: Image-mode volumes are special volumes that have a direct relationship with one MDisk. If you have a MDisk that contains data that you want to merge into the clustered system, you can create an image-mode volume.

**Easy Tier function**
SAN Volume Controller includes Storage Easy Tier, a function that responds to the presence of solid-state drives (SSDs) in a storage pool that also contains hard disk drives (HDDs). The system automatically and non-disruptively moves frequently accessed data from HDD MDisks to SSD MDisks, placing the data in a faster tier of storage.

In this dynamically tiered environment, data movement is seamless to the host application regardless of the storage tier in which the data is located.

**Quorum disks**
Quorum disks are used when there is a problem in the SAN fabric or when nodes are shut down. SAN Volume Controller uses quorum disks in such situation to ensure data consistency and data integrity while maintaining data access. A quorum disk determines which group of nodes stops operating and processing I/O requests. In this tie-break situation, the first group of nodes that accesses the quorum disk marks its ownership of the quorum disk. As a result, the group continues to operate as the system, handling all I/O requests.

**Access modes**
The access mode determines how the SAN Volume Controller system uses the MDisk. The following are the types of access modes:
- Unmanaged: The MDisk is not used by the system.
- Managed: The MDisk is assigned to a storage pool and provides extents that volumes can use.
- Image: The MDisk is assigned directly to a volume with a one-to-one mapping of extents between the MDisk and the volume.
- Array: The MDisk represents a set of drives in a RAID from internal storage.

**Important:** If you add an MDisk that contains existing data to a storage pool while the MDisk is in unmanaged or managed mode, the data is lost. The image mode is the only mode that preserves this data.

**Mirrored volumes**
Volume mirroring allows a volume to have two physical copies. The mirrored volume feature provides a simple RAID-1 function. Each volume copy can belong to a different storage pool, and can belong to separate storage subsystems.

The primary copy indicates the preferred volume for read requests. When a server writes to a mirrored volume, the system writes the data to both copies. You can create a volume with one or two copies, and you can convert a non-mirrored volume into a mirrored volume by adding a copy. Alternately, you can convert a mirrored volume into a non-mirrored volume by deleting one copy or by splitting one copy to create a new, non-mirrored volume.

**System state**
The state of the clustered system holds all of the configuration and internal data. The system state information is held in nonvolatile memory. If the power fails, the UPS units maintain internal power long enough for system state information to be stored. This information is stored on the SAN Volume Controller internal disk drive of each node.

**Back up and restore process**
Planning your backup/restore procedures is an important part of a complete backup and disaster recovery solution.

Backup is the process of extracting configuration settings from a SAN Volume Controller system and writing them to disk. The restore process uses backup configuration data files to restore system configuration.

If power fails on a system or a node in a system is replaced, the system configuration settings are automatically restored. This restoration occurs when the repaired node is added to the system. To restore the system configuration in a disaster, plan to back up the system configuration settings to tertiary storage. Use the configuration backup functions to back up the system configuration.

**Important:** For complete disaster recovery, regularly back up the business data that is stored on volumes at the application server or host level.

**Event notifications**
SAN Volume Controller can use Simple Network Management Protocol (SNMP) traps, syslog messages, and call home email. These alerts notify you and the IBM Support Center when significant events are detected.

These notification methods can be used simultaneously. Notifications are normally sent immediately after an event occurs.
SAN Volume Controller migration

The SAN Volume Controller allows you to migrate extents that belong to a volume from source managed disk (MDisk) to a target MDisk. It can do so without interrupting host access to the associated volume. This function is used when performing volume migrations, and can be performed for any volume defined on the SAN Volume Controller.

SAN Volume Controller migration can be used for the following tasks:

- Redistribution of volumes and their workload within an SAN Volume Controller cluster across back-end storage
- Moving workload onto newly installed storage subsystems
- Moving workload off storage so that old or failing storage subsystems can be decommissioned
- Moving workload to rebalance a changed workload
- Migrating data from earlier model back-end storage to SVC-managed storage
- Migrating data from one back-end controller to another using the SAN Volume Controller as a data block mover
- Removing the SAN Volume Controller from the SAN
- Migrating data from managed mode back into image mode before removing the SAN Volume Controller from a SAN

SAN Volume Controller migration can be performed at either the volume or the extent level, depending on the purpose of the migration. The following are the supported migration activities:

- Migrating extents within a storage pool and redistributing the extents of a volume on the MDisks in the storage pool
- Migrating extents off an MDisk to other MDisks in the storage pool so the MDisk can be removed
- Migrating a volume from one storage pool to another
- Changing the virtualization type of the volume to Image mode
- Migrating a volume between I/O groups

Extents

An extent is a fixed-size unit of data that is used to manage the mapping of data between MDisks and volumes. The extent size choices are 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, and 8192 MB. The choice of extent size affects the total storage that can be managed by the SAN Volume Controller. For the 8192-MB extent size, SAN Volume Controller supports the following capacities:

- A maximum volume capacity of 256 TB
- A maximum SAN Volume Controller system capacity of 32 PB
Figure 7-1 shows the relationship between extents and a volume.

![Figure 7-1 Extents used to create a volume](image)

Figure 7-2 shows the relationship between physical and virtual disks.

![Figure 7-2 Relationships between physical and virtual disks](image)

**Consistency groups**

Consistency groups preserve data consistency across multiple volumes, especially for applications that have related data that spans multiple volumes. To preserve the integrity of the data being written, ensure that dependent writes are run in the intended sequence of the application.

**Multipath Subsystem Device Driver (SDD)**

For resilience, availability, redundancy, and throughput reasons, use multipath device drivers. IBM Multipath Subsystem Device Driver (SDD) is available for most operating system platforms. In addition, some vendors provide their own.
See the IBM SAN Volume Controller website for the latest device drivers and the support matrix at:

http://www.ibm.com/storage/support/2145

SDD software for all supported platforms can be obtained for no additional fee at:

https://www-304.ibm.com/support/docview.wss?uid=ssg1S7001350

### 7.1.2 SAN Volume Controller copy services

The advanced function copy services offered by the SAN Volume Controller are FlashCopy, Metro Mirror, and Global Mirror. If you plan to use SAN Volume Controller for all your copy services needs, you might not need to purchase additional premium copy services features.

The SAN Volume Controller copy services functions can provide additional capabilities and unique advantages over other storage devices:

- SAN Volume Controller supports consistency groups for FlashCopy, Metro Mirror, and Global Mirror.
- Consistency groups in SAN Volume Controller can span across underlying storage subsystems.
- FlashCopy source volumes that are on one disk system can write to target volumes on another disk system.
- SAN Volume Controller supports FlashCopy targets as Metro or Global Mirror sources (SAN Volume Controller code 6.2.0.1)
- Metro Mirror and Global Mirror source volumes can be copied to target volumes on a dissimilar storage subsystem.

### 7.1.3 Metro Mirror

The Metro Mirror copy service is a synchronous remote copy function. Metro Mirror in SAN Volume Controller is similar to Metro Mirror in the IBM System Storage DS® family.

The function of Metro Mirror is to maintain two real-time synchronized copies of a data set. Often, the two copies are geographically dispersed on two SAN Volume Controller clusters, although you can use Metro Mirror in a single cluster (within an I/O group). If the primary copy fails, the secondary copy can then be enabled for I/O operation.

Metro Mirror works by defining a Metro Mirror relationship between volumes of equal size. When creating the Metro Mirror relationship, define one volume as the master, and the other volume as the auxiliary. Any data that exists on the auxiliary volume before the relationship being set up is deleted.

The terms *master* and *auxiliary* are used instead of the industry standard terms of source and target. These terms are used because they match the parameter keywords used in the commands to set up the relationships between the volumes. The following terms are all interchangeable:

- Source and target
- Primary and secondary
- Master and auxiliary

To provide management and data consistency across a number of Metro Mirror relationships, consistency groups are supported.
The SAN Volume Controller provides both intracluster and intercluster Metro Mirror as described in the following sections. The examples of migration in this chapter use two SAN Volume Controller clusters, which is intercluster Metro Mirroring.

**Intracluster Metro Mirror**
Intracluster Metro Mirror can be applied within any single I/O group. Metro Mirror across I/O groups in the same SAN Volume Controller cluster is not supported.

**Intercluster Metro Mirror**
A Metro Mirror session is a form of synchronous remote replication, designed to operate over distances under 300 kilometers (km). It maintains identical data in both the source and target. Changes made to the source data are propagated to the target before the write finishes posting. With Metro Mirror, the source is located in one subsystem and the target is located in another subsystem.

Intercluster Metro Mirror operations require a pair of SAN Volume Controller clusters that are separated by a number of moderately high-bandwidth links. The two SAN Volume Controller clusters must be defined in a SAN Volume Controller partnership. This definition must be performed on both SAN Volume Controller clusters to establish a fully functional Metro Mirror partnership. Correct link sizing is crucial to successfully implement Metro Mirror replication.

Metro Mirror is a fully synchronous remote copy technique. It ensures that updates (writes), are committed at both primary and secondary volumes before the application is given “write successful” status to an update and released. A write to the master volume is mirrored to the cache for the auxiliary volume. An acknowledge of the write is then sent back to the host (Figure 7-3).

![Remote Mirroring synchronous write sequence](image)

If your system goes down, Metro Mirror provides nearly zero loss if data must be used from the recovery site. While the Metro Mirror relationship is active, the secondary copy (auxiliary volume) is not accessible for host application write I/O at any time. The SAN Volume Controller allows read-only access to the secondary volume, when it contains a consistent image. To allow access to the secondary volume for host operations, the Metro Mirror relationship must first be stopped.
7.1.4 Global Mirror

Global Mirror (GM) works by defining a GM relationship between two volumes of equal size and maintains the data consistency in an asynchronous manner.

Because SAN Volume Controller Global Mirror is not intended to use for data migration, it is not addressed further here. For more information about SAN Volume Controller Global Mirror, see Software Installation and Configuration Guide, GC27-2286-01.

7.2 SAN Volume Controller concepts for migrating the data

This section explains how to use SAN Volume Controller as a migration tool for block-level (LUN) migrations between storage subsystems. SAN Volume Controller is a robust and reliable data migration solution. When addressing complexity and data integrity, SAN Volume Controller is usually the best solution for data migration. The following SAN Volume Controller methods are available to use in migration scenarios:

- Data migration using SAN Volume Controller volume migration
- Data migration using SAN Volume Controller FlashCopy
- Data migration using SAN Volume Controller Metro Mirror
- Data migration using mirrored volumes

7.2.1 Data migration using SAN Volume Controller volume migration

SAN Volume Controller Volume migration combines migrating a volume to an image mode volume, with the ability to migrate between storage pools. The source for the migration is an image mode VDisk.

To be able to migrate, the destination MDisk must be greater than or equal to the size of the VDisk. Also, the MDisk specified as the target must be in an unmanaged state at the time the command is run.

If the migration is interrupted by a cluster recovery, the migration will resume after the recovery completes.

Volume migration algorithm

The following section describes the algorithm for the volume migration.

Regardless of the extent size for the Storage pool, data is migrated in units of 16 MB. In this description, this unit is called a chunk.

The algorithm to migrate an extent includes the following steps:

1. Pause all I/O on the source MDisk on all nodes in the SAN Volume Controller cluster. During the pause, the specific mdisk SVC stores all I/O in the virtualization layer and waits for all outstanding requests to complete. Therefore, the host is not aware that the migration is underway. The I/O to other extents is unaffected.

2. Unpause I/O on all of the source MDisk extents other than writes to the specific chunk that is being migrated. Writes to the extent are mirrored to the source and destination.

3. On the node performing the migration, for each 256K section of the chunk:
   - Synchronously read 256K from the source.
   - Synchronously write 256K to the target.
4. After the entire chunk is copied to the destination, repeat the process for the next chunk in the extent.

5. After the entire extent is migrated, perform the following steps:
   a. Pause all I/O to the extent being migrated
   b. Checkpoint the extent move to on-disk metadata
   c. Redirect all further reads to the destination
   d. Stop mirroring writes (writes only to the destination)
   If the checkpoint fails, the I/O is unpaused.

During the migration, the extent can be divided into three regions as shown in Figure 7-4. Region B is the chunk that is being copied. The reads and writes are in the following states:

- Writes to region B are queued (paused) in the virtualization layer, waiting for the chunk to be copied.
- Reads to Region A are directed to the destination because this data has already been copied.
- Writes to Region A are written to both the source and the destination extent to maintain the integrity of the source extent.
- Reads and writes to Region C are directed to the source because this region has yet to be migrated.

The migration of a chunk requires 64 synchronous reads and 64 synchronous writes. During this time, all writes to the chunk from higher layers in the software stack (such as cache destages) are held back. If the back-end storage is operating with significant latency, this operation might take some time to complete. This latency can have an adverse effect on the overall performance of the SAN Volume Controller. To avoid this situation, if the migration of a particular chunk is still active after one minute, the migration is paused for 30 seconds. During this time, writes to the chunk are allowed to proceed. After 30 seconds, the migration of the chunk is resumed. This process is repeated as many times as necessary to complete the migration of the chunk.
Considerations for host I/O during volume migration
SAN Volume Controller guarantees read stability during data migrations, even if the data migration is stopped by a node reset or a cluster shutdown. SAN Volume Controller disallows writes on all nodes to the area being copied, and upon a failure the extent migration is restarted from the beginning.

However, SAN Volume Controller does not guarantee data synchronization on writes during the migration. After the metadata is updated for each 16-MB chuck, any new writes are redirected to the target disk only. Because of the nature of the migration algorithm, restarting hosts in the event of an SAN Volume Controller cluster failure is difficult.

Important: To guarantee that the source and target data is synchronous using this method of migration, pause I/O during the migration. The risk of a complete SAN Volume Controller cluster failure during migration is small. Weigh the decision to pause I/O during migration against the impact of any I/O downtime.

7.2.2 Data migration using SAN Volume Controller FlashCopy

The FlashCopy function allows you to create a rapid, complete, and consistent copy from a source volume to a target volume. This feature can be used effectively for migrating data to and from SAN Volume Controller managed storage.

When SAN Volume Controller is introduced, it takes time and effort to virtualize the storage. Migration of data from non-virtualized to virtualized storage can be achieved in many different ways. Image to managed volume migration and volume mirroring are the traditional approaches for data migration.

FlashCopy can also be used for migrating data from non-virtualized to SAN Volume Controller managed storage. The target volume can be anywhere within the SVC-managed environment. The typical scenario for the migration is from image volume to managed volume. FlashCopy allows you to create a read/write copy on the target volume in a matter of a few seconds.

The significant advantage of FlashCopy migration is that the source volume remains unchanged throughout the migration process. Another advantage is that when the copy is created, the target volume can instantly be used in read/write mode.

For details and information about SAN Volume Controller Copy Services, see SAN Volume Controller V4.3.0 Advanced Copy Services, SG24-7574.

FlashCopy-based data migration provides a simple and reliable backout path. If something did not go as planned during the migration, unmask the original volumes from the SAN Volume Controller and mask them back to a server. You do not need to copy data back because the original volumes remained unchanged.

You can also repeat the copy process from the source to the target as many times as required. If for whatever reason the target set of volumes become unusable, you can refresh the contents by recopying data.

The disadvantage of this method is that it requires slightly more effort compared to the volume migration and volume mirroring methods. The reason for this is you need to create the target managed volumes of matching sizes, and create and start one FlashCopy mapping per volume pair. However, this process can be automated using migration scripts.
A FlashCopy based data migration process from non-virtualized to SAN Volume Controller managed storage involves the following steps:

1. Stop applications on the server.
2. Uninstall the multipathing device driver.
3. Shut down the server.
4. Remove volume masking to the server.
5. Present volumes to SAN Volume Controller.
6. Discover MDisks on SAN Volume Controller.
7. Create Image mode volumes using the discovered MDisks.
8. Create matching size managed volumes.
9. Create FlashCopy mappings between source and target volumes.
10. Start the FlashCopy copying process.
11. Create and activate a zone between the server and SAN Volume Controller.
12. Create host mapping to present target volumes to the server.
13. Start the server.
14. Install the multipathing device driver.
15. Perform data validation for OS and applications.
16. Perform cleanup.

The FlashCopy mappings in step 9 can be created with `-autodelete` flag so that the mapping is removed after background copy process is finished. By removing the mapping automatically, you reduce the number of steps in the cleanup process in step 15.

After the FlashCopy mapping is created and the background copy process is started, monitor the progress and adjust the copy rate to minimize the performance impact. In most cases, the background copy has no measurable performance impact on the application workload. In this case, you can adjust the rate so that the background copy can complete sooner. You can schedule copy rate reduction, for example to minimize the performance impact during a period of heavy batch processing.

In order for a data migration to be successful the set of volumes must be consistent. To ensure volume consistency, all the volume manipulations must be done for the entire volume set. You must not present a subset of volumes to be accessed by a server.

### 7.2.3 Data migration using SAN Volume Controller Metro Mirror

Data migrations using Metro Mirror are more complicated. Use two SAN Volume Controller clusters (four nodes: each cluster must have a minimum of two nodes) when migrating using Metro Mirror. You can use a single cluster Metro Mirror (intrACLuster Metro Mirror) for data migration, but this configuration introduces a single point of failure. Any problems occurring in this single SAN Volume Controller cluster or the fabric connecting the environment might affect both the source and target data. This risk is greater before the source and target are fully synchronized.

By using dual cluster Metro Mirroring (intercluster Metro Mirror), the source volumes receive the host updates at all times during the data migration. If a problem occurs between the host and its data volumes, a normal recovery is possible. In the background, the SAN Volume Controller copies the data over to the target volumes. You can restart from the source or, if necessary, from the target volumes.

The target LUN must be exactly the same size as the source LUN.
7.2.4 Data migration using mirrored volumes

Using volume mirroring creates two physical copies of a volume. Each volume copy can belong to a separate storage pool, and each copy has the same virtual capacity as the volume. In the management GUI, an asterisk (*) indicates the primary copy of the mirrored volume. The primary copy indicates the preferred volume for read requests.

When a server writes to a mirrored volume, the system writes the data to both copies. When a server reads a mirrored volume, the system picks one of the copies to read. If one of the mirrored volume copies is temporarily unavailable, the volume remains accessible to servers through the other copy. The system remembers which areas of the volume are written and resynchronizes these areas when both copies are available.

You can create a volume with one or two copies, and you can convert a non-mirrored volume into a mirrored volume by adding a copy. When a copy is added in this way, the SAN Volume Controller clustered system synchronizes the new copy so that it is the same as the existing volume. Servers can access the volume during this synchronization process.

You can convert a mirrored volume into a non-mirrored volume by deleting one copy or by splitting one copy to create a new non-mirrored volume. The volume copy can be any type: Image, striped, sequential, and either thin provisioned or fully allocated. The two copies can be of different types.

You can use mirrored volumes for the following reasons:

- Improving availability of volumes by protecting them from a single storage system failure.
- Providing concurrent maintenance of a storage system that does not natively support concurrent maintenance.
- Providing an alternative method of data migration with better availability characteristics. While a volume is being migrated using the data migration feature, it is vulnerable to failures on both the source and target storage pool. Volume mirroring starts with a non-mirrored volume in the source storage pool, and then adds a copy to that volume in the destination storage pool. When the volume is synchronized, you can delete the original copy. During the synchronization process, the volume remains available even if there is a problem with the destination storage pool.
- Converting between fully allocated volumes and thin-provisioned volumes.

When you use volume mirroring, consider how quorum candidate disks are allocated. Volume mirroring maintains some state data on the quorum disks. If a quorum disk is not accessible and volume mirroring is unable to update the state information, a mirrored volume might need to be taken offline to maintain data integrity. To ensure the high availability of the system, ensure that multiple quorum candidate disks, allocated on different storage systems, are configured.
Figure 7-5 shows a logical layout of mirrored volumes.

![Diagram of mirrored volumes](image)

**Figure 7-5   SAN Volume Controller mirrored volumes**

### 7.3 Migrating using SAN Volume Controller

This section explains how to migrate from a conventional storage infrastructure to a virtualized storage infrastructure using the IBM System Storage SAN Volume Controller (SAN Volume Controller). The section includes migrating between different types of MDisks. It also addresses how to phase SAN Volume Controller out of a virtualized storage infrastructure. You might want to phase it out, for example, after a trial period or after using the SAN Volume Controller as a data migration tool. It includes examples of using Metro Mirror to migrate data, and how to use SAN Volume Controller mirrored volume functions to migrate data from one disk subsystem to another.

You can perform migration at either the volume or the extent level, depending on the purpose of the migration. The following migration activities are supported:

- Migrating extents within a storage pool, redistributing the extents of a volume on the MDisks within the same storage pool
- Migrating extents off an MDisk, which is removed from the storage pool, to other MDisks in the same storage pool
- Migrating a volume from one storage pool to another
- Migrating a volume to change the virtualization type of the volume to image
- Migrating a volume between I/O groups
7.3.1 Migrating extents

The SAN Volume Controller provides various data migration features. These features can be used to move data both within storage pools and between storage pools. These features can be used concurrently with I/O operations. You can use either of these methods to migrate data:

1. Migrating data (extents) from one MDisk to another within the same storage pool. This method can be used to move data off highly used MDisks.
2. Migrating volumes from one storage pool to another. This method can be used to move data off highly used storage pools.

You can determine which MDisks are heavily used by gathering and analyzing input/output (I/O) statistics about nodes, MDisks, and volumes. Using this information, you can migrate extents to less used MDisks in the same storage pool. This migration can only be performed using the command-line tools.

If performance monitoring tools indicate that a managed disk in the pool is being overused, migrate some of the data to other MDisks within the same storage pool.

1. Determine the number of extents that are in use by each volume for the MDisk using the following CLI command:
   lsmdiskextent mdiskname
2. From the number of extents that each volume is using on the MDisk, select some of them to migrate elsewhere in the group.
3. Determine the storage pool that the MDisk belongs to using this CLI command:
   lsmdisk mdiskname | ID
4. List the MDisks in the group by issuing the following CLI command:
   lsmdisk -filtervalue mdisk_grp_name=mdiskgrpname
5. Select one of these MDisks as the target MDisk for the extents. You can determine how many free extents exist on an MDisk using the CLI command:
   lsfreeextents mdiskname
6. Issue the lsmdiskextent newmdiskname command for each of the target MDisks to ensure that you are not just moving the over-utilization to another MDisk. Check that the volume that owns the set of extents to be moved does not already own a large set of extents on the target MDisk.
7. For each set of extents, issue this CLI command to move them to another MDisk:
   migrateexts -source mdiskname | ID -exts \[num_extents\] -target newmdiskname | ID -threads 4 vdiskid
   Where [num_extents] is the number of extents on the vdiskid. The newmdiskname | ID value is the name or ID of the MDisk to migrate this set of extents to.

   **Tip:** The number of threads indicates the priority of the migration processing, where 1 is the lowest priority and 4 is the highest priority.

8. Repeat the previous steps for each set of extents that you are moving.

You can check the progress of the migration by issuing this CLI command:
lsmigrate
7.3.2 Migrating extents off an MDisk that is being deleted

Before deleting an MDisk from a storage pool using the `rmmdisk -force` command, migrate the extents on it onto other MDisks in the storage pool.

If a volume uses extents that need to be moved as a result of a `rmmdisk` command, the virtualization type for that volume must be set to striped. This process is needed only if it was previously sequential or image.

If the MDisk is operating in image mode, the MDisk transitions to managed mode while the extents are being migrated. Upon deletion, it changes to unmanaged mode.

**Remember:** If the `-force` flag is not used and volumes occupy extents on one or more of the MDisks that are specified, the command fails.

When the `-force` flag is used and volumes occupy extents on the specified MDisks, all extents are migrated to other MDisks in the storage pool. This process will occur only if there are enough free extents in the storage pool. The deletion of the MDisks is postponed until all extents are migrated, which can take some time. If there are insufficient free extents in the storage pool, the command fails.

7.3.3 Migrating a volume between storage pools

An entire volume can be migrated from one storage pool to another storage pool using the `migratevdisk` command. A volume can be migrated between storage pools regardless of the virtualization type (image, striped, or sequential), although it changes to striped. The command you need to use varies depending on the type of migration, as shown in Table 7-2.

<table>
<thead>
<tr>
<th>Storage pool-to-storage pool type</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Managed to managed</td>
<td><code>migratevdisk</code></td>
</tr>
<tr>
<td>Image to managed</td>
<td><code>migratevdisk</code></td>
</tr>
<tr>
<td>Managed to image</td>
<td><code>migratetoimage</code></td>
</tr>
<tr>
<td>Image to image</td>
<td><code>migratetoimage</code></td>
</tr>
</tbody>
</table>

**Rule:** The source and destination storage pool must have the same extent size for migration to take place. Volume mirroring can also be used to migrate a volume between storage pools if the extent sizes of the two pools are not the same.

Migration commands fail if the target or source volume is offline, or if there is insufficient quorum disk space to store the metadata. Correct the offline or quorum disk condition and reissue the command.

You can migrate volumes between storage pools using the command-line interface (CLI).

Determine the usage of particular MDisks by gathering input/output (I/O) statistics about nodes, MDisks, and volumes. Analyze it to determine which volumes or MDisks need to be moved to another storage pool.
Perform the following step to gather statistics about MDisks and volumes:

1. Use secure copy (`scp`) command to retrieve the dump files for analyzing. For example, issue the following command, which copies all the volume statistics files to the AIX host in the current directory:
   
   ```
   scp clusterip:/dumps/iostats/v_*
   ```

2. Analyze the memory dumps to determine which volumes are overused. Determine which MDisks are being used heavily so you can spread the data more evenly across all MDisks in the group.

After you analyze the I/O statistics data, you can determine which volumes are hot. You also need to determine the storage pool that you want to move this volume to. Either create a storage pool or determine an existing group that is not yet heavily used. Check the I/O statistics files that you generated for heavily used groups. Ensure that the MDisks or VDisks in the target storage pool are used less than those in the source group.

You can use data migration or volume mirroring to migrate data between MDisk groups. Data migration uses the command `migratevdisk`. Volume mirroring uses the commands `addvdiskcopy` and `rmvdiskcopy`.

When you issue the `migratevdisk` command, a check is made to ensure that the destination of the migration has enough free extents to satisfy the command. If it does not, the command fails. This command takes some time to complete.

Perform the following steps to use the `migratevdisk` command to migrate volumes between storage pools:

1. After you determine the volume that you want to migrate and the new storage pool you want to migrate it to, issue the following CLI command:
   
   ```
   migratevdisk -vdisk vdiskname/ID -mdiskgrp newmdiskgrname/ID -threads 4
   ```

2. You can check the progress of the migration by issuing the following CLI command:
   
   ```
   lsmigrate
   ```

**Note:** When you use data migration, the volume goes offline if either storage pool fails. Volume mirroring can be used to minimize the impact to the volume because the volume goes offline only if the source storage pool fails.

### 7.3.4 Using volume mirroring

Perform the following steps to use volume mirroring to migrate volumes between storage pools:

1. After you determine the volume that you want to migrate and the new storage pool that you want to migrate it to, issue the following command:
   
   ```
   addvdiskcopy -mdiskgrp newmdiskgrname/ID vdiskname/ID
   ```

2. The copy ID of the new copy is returned. The copies now synchronize so the data is stored in both storage pools.

3. You can check the progress of the synchronization by issuing the following command:
   
   ```
   lsvdisksyncprogress
   ```
4. After the synchronization is complete, remove the copy from the original I/O group to free up extents and decrease the utilization of the storage pool. To remove the original copy, issue the following command:

\texttt{rmvdiskcopy -copy original copy id vdiskname/ID}

### 7.3.5 Image mode volume migration

This section addresses migrating data from an image mode volume to a fully managed volume. This type of migration is used to take an existing host LUN and move it into the virtualization environment as provided by the SAN Volume Controller.

#### MDisk modes

There are three MDisk modes:

- **Unmanaged MDisk**: An MDisk is reported as unmanaged when it is not a member of any storage pool. An unmanaged MDisk is not associated with any volumes and has no metadata stored on it. The SAN Volume Controller does not write to an MDisk that is in unmanaged mode. The only exception is when it attempts to change the mode of the MDisk to one of the other modes.

- **Image mode MDisk**: Image mode provides a direct block-for-block translation from the MDisk to the volume with no virtualization. Image mode volumes have a minimum size of one block (512 bytes) and always occupy at least one extent. An image mode MDisk is associated with exactly one volume.

- **Managed mode MDisk**: Managed mode MDisks contribute extents to the pool of available extents in the storage pool. Zero or more managed mode volumes might use these extents.

#### Changing between the modes

The following state transitions can occur to an MDisk (Figure 7-6 on page 195):

- **Unmanaged mode to managed mode**: This transition occurs when an MDisk is added to a storage pool, which makes the MDisk eligible for the allocation of data and metadata extents.

- **Managed mode to unmanaged mode**: This transition occurs when an MDisk is removed from a storage pool.

- **Unmanaged mode to image mode**: This transition occurs when an image mode MDisk is created on an MDisk that was previously unmanaged. It also occurs when an MDisk is used as the target for a migration to image mode.

- **Image mode to unmanaged mode**: There are two ways in which this transition can happen:
  - When an image mode volume is deleted, the MDisk that supported the volume becomes unmanaged.
  - When an image mode volume is migrated in image mode to another MDisk, the MDisk that is being migrated from remains in image mode. When all data is moved off the MDisk, it changes to unmanaged mode.
- Image mode to managed mode: This transition occurs when the image mode volume that is using the MDisk is migrated into managed mode.
- Managed mode to image mode: There is no operation that takes an MDisk directly from managed mode to image mode. You can achieve this transition by performing operations that convert the MDisk to unmanaged mode and then to image mode.

Image mode volumes have the special property that the last extent in the volume can be a partial extent. Managed mode disks do not have this property.

To perform any type of migration activity on an image mode volume, the image mode disk must first be converted into a managed mode disk. If the image mode disk has a partial last extent, this last extent in the image mode volume must be the first extent to be migrated. This migration is handled as a special case. After the special migration operation occurs, the volume becomes a managed mode volume. If the image mode disk does not have a partial last extent, no special processing is needed. The image mode volume is changed into a managed mode volume and is treated in the same way as any other managed mode volume.

After data is migrated off a partial extent, there is no way to migrate data back onto the partial extent.

You can use the command-line interface (CLI) to import storage that contains existing data and continue to use this storage. You can also use the advanced functions, such as Copy Services, data migration, and the cache. These disks are known as image mode virtual volumes.

Make sure that you are aware of the following restrictions before you create image mode volumes:

1. Unmanaged-mode managed disks (MDisks) that contain existing data cannot be differentiated from unmanaged-mode MDisks that are blank. Therefore, it is vital that you control the introduction of these MDisks to the clustered system by adding these disks one at a time.

2. Do not manually add an unmanaged-mode MDisk that contains existing data to a storage pool. If you do, the data will be lost. Use the command to convert an image mode volume from an unmanaged-mode disk, and select the storage pool you want it added to.
Perform the following steps to create an image mode volume:

1. Stop all I/O operations from the hosts.
2. Unmap the logical disks that contain the data from the hosts.
3. Create one or more storage pools.
4. Map a single array or logical unit from your RAID storage system to the clustered system.
   You can do this mapping through a switch zoning or a RAID storage system based on your host mappings. The array or logical unit is displayed as an unmanaged-mode MDisk to the SAN Volume Controller.
5. Issue the `lsmdisk` command to list the unmanaged-mode MDisks. If the new unmanaged-mode MDisk is not listed, perform a fabric-level discovery. Issue the `detectmdisk` command to scan the Fibre Channel network for the unmanaged-mode MDisks.

**Tip:** The `detectmdisk` command also rebalances MDisk access across the available storage system device ports.

6. Convert the unmanaged-mode MDisk to an image mode virtual disk. Issue the `mkvdisk` command to create an image mode virtual disk object.
7. Map the new volume to the hosts that were previously using the data that the MDisk now contains.

You can use the `mkvdiskhostmap` command to create a mapping between a volume and a host. This mapping makes the image mode volume accessible for I/O operations to the host.

After the volume is mapped to a host object, the volume is detected as a disk drive with which the host can perform I/O operations.

If you want to virtualize the storage on an image mode volume, you can transform it into a striped volume. Migrate the data on the image mode volume to managed-mode disks in another storage pool. Issue the `migratevdisk` command to migrate an entire image mode volume from one storage pool to another storage pool.

### 7.3.6 Migrating the volume to image mode

You can migrate a volume to an image mode volume and migrate between storage pools at the same time. The source for the migration can be a managed mode or an image mode volume. This leads to four possibilities:

- Migrate image mode-to-image mode within a storage pool.
- Migrate managed mode-to-image mode within a storage pool.
- Migrate image mode-to-image mode between storage pools.
- Migrate managed mode-to-image mode between storage pools.

The following conditions must apply before you can migrate:

- The destination MDisk must be greater than or equal to the size of the volume.
- The MDisk that is specified as the target must be in an unmanaged state at the time that the command is run.

If the migration is interrupted by a cluster recovery, the migration will resume after the recovery completes.
If the migration involves moving between storage pools, the volume behaves as described in 7.3.3, “Migrating a volume between storage pools” on page 192.

Regardless of the mode in which the volume starts, it is reported as being in managed mode during the migration. In addition, both of the MDisks involved are reported as being in image mode during the migration. Upon completion of the command, the volume is classified as an image mode volume. Issuing this command results in the inclusion of the MDisk into the user specified MDisk group.

The `migratetoimage` CLI command allows you to migrate the data from an existing VDisk (volume) onto a managed disk (MDisk). When it is issued, it migrates the data of the user specified source VDisk onto the specified target MDisk. When the command completes, the VDisk is classified as an image mode VDisk.

**Note:** Migration commands fail if the target or source VDisk is offline, or if there is insufficient quorum disk space to store the metadata. Correct the offline or quorum disk condition and reissue the command.

Issue the following CLI command to migrate data to an image mode VDisk:

```
```

Where:

- `[vdiskname/ID]` is the name or ID of the VDisk
- `[newmdiskname/ID]` is the name or ID of the new MDisk
- `[newmdiskgrpname/ID]` is the name or ID of the new MDisk group (storage pool)

### 7.3.7 Migrating a volume between I/O groups

A volume can be migrated between I/O groups by using the `svctask chvdisk` command. This command is only supported if the volume is not in a FlashCopy Mapping or Remote Copy relationship.

To move a volume between I/O groups, the cache must first be flushed. The SAN Volume Controller attempts to destage all write data for the volume from the cache during the I/O group move. This flush fails if data is pinned in the cache for any reason, such as a storage pool being offline. By default, this failed flush causes the migration between I/O groups to fail, but this behavior can be overridden using the `-force` flag. If the `-force` flag is used and SAN Volume Controller cannot destage all write data from the cache, the cached data is lost. During the flush, the volume operates in cache write-through mode.

**Important:** Do not move a volume to an offline I/O group under any circumstance. Ensure that the I/O group is online before you move the volumes to avoid any data loss.

You must quiesce host I/O before the migration for two reasons:

- If there is significant data in cache that takes a long time to destage, the command-line times out.
- Subsystem Device Driver vpaths that are associated with the volume are deleted before the volume move takes place to avoid data corruption. Therefore, data corruption can occur if I/O is still occurring for a particular LUN ID.

When migrating a volume between I/O groups, you can specify the preferred node, or you can allow SAN Volume Controller assign the preferred node.
A volume that is a member of a FlashCopy mapping or a Remote Copy relationship cannot be moved to another I/O group. You cannot override this restriction using the \texttt{-force} flag. You must delete the mapping or relationship before the volume can be migrated between I/O groups.

Modifying the I/O group that services the volume cannot be done concurrently with I/O operations. It also requires a rescan at the host level to ensure that the multipathing driver is notified of the following conditions:

- The allocation of the preferred node has changed
- The ports by which the volume is accessed have changed.

Modify the group only when one pair of nodes becomes overused.

Perform the following steps to migrate a volume between I/O groups:

1. Synchronize all file systems that are mounted on the volume.
2. Stop all I/O operations to the volume.
3. Issue the following CLI command to migrate the volume into a new I/O group:

   \texttt{chvdisk -iogrp [iogrp_name_or_id] -node [preferred_node] [vdisk]}

   Where:
   - \texttt{[iogrp_name_or_id]} is the name or ID of the I/O group that you want to migrate the volume to
   - \texttt{[preferred_node]} is the name of the node that you want to move the volume to
   - \texttt{[vdisk]} is the name of the volume that you want to migrate
4. Resynchronize the volume to host mapping. For more information, see the \textit{IBM System Storage Multipath Subsystem Device Driver User's Guide} or the documentation provided with your multipathing driver.
5. Restart the I/O operations to the volume.

\textbf{7.3.8 Monitoring the migration progress}

To monitor the progress of ongoing migrations, use the \texttt{svcinfo lsmigrate} CLI command. To determine the extent allocation of MDisks and volumes, use the following commands.

- To list the volume IDs and the corresponding number of extents that the volumes occupy on the queried MDisk, use the following CLI command:
  \texttt{svcinfo lsmdiskextent <mdiskname | mdisk_id>}
- To list the MDisk IDs and the corresponding number of extents that the queried volumes occupy on the listed MDisks, use the following CLI command:
  \texttt{svcinfo lsvdiskextent <vdiskname | vdisk_id>}
- To list the number of available free extents on an MDisk, use the following CLI command:
  \texttt{svcinfo lsfreeextents <mdiskname | mdisk_id>}

\textbf{Important:} After a migration is started, you cannot stop it. The migration runs to completion unless it is stopped or suspended by an error condition, or the volume being migrated is deleted.

If you want to start, suspend, or cancel a migration, or control the rate of migration, use the volume mirroring function or migrate volumes between storage pools.
7.3.9 Parallelism

You can perform the following activities in parallel. The activities are divided into the following categories:

- Per cluster
- Per MDisk

**Per cluster**

An SAN Volume Controller cluster supports up to 32 active concurrent instances of the following migration activities:

- Migrate multiple extents
- Migrate between storage pools
- Migrate off a deleted MDisk
- Migrate to image mode

These high-level migration tasks can be started by scheduling single extent migrations. Up to 256 single extent migrations can run concurrently. This number can be any combination of the migration activities previously listed.

The `migrate multiple extents` and `migrate between storage pools` commands support a flag that allows you to specify the number of parallel “threads” to use. The flag can be set from 1 - 4. This parameter affects the number of extents that are concurrently migrated for that migration operation. Thus, if the thread value is set to 4, up to four extents can be migrated concurrently for that operation, subject to other resource constraints.

**Per MDisk**

The SAN Volume Controller supports up to four concurrent single extent migrates per MDisk. This limit does not take into account whether the MDisk is the source or the destination. If more than four single extent migrates are scheduled for a particular MDisk, further migrations are queued pending the completion of the currently running migrations.

7.3.10 Error handling

The migration is suspended if any of the following conditions exist. If any other error is encountered, it is stopped:

- The migration is between storage pools and has progressed beyond the first extent: These migrations are always suspended because stopping a migration in progress leaves a volume spanning storage pools. Spanning storage pools is not a valid configuration other than during a migration.
- The migration is a Migrate to Image Mode, even if it is processing the first extent: These migrations are always suspended rather than stopped because stopping a migration in progress leaves the volume in an inconsistent state.
- A migration is waiting for a metadata checkpoint that has failed.

If a migration is stopped and migrations are queued awaiting the use of the MDisk, these migrations now commence. However, if a migration is suspended, the migration continues to use resources, and another migration is not started.

The SAN Volume Controller attempts to resume the migration if the error log entry is marked as fixed using the CLI or the GUI. If the error condition no longer exists, the migration proceeds. The migration might resume on a node other than the node that started the migration.
7.3.11 Migration tips

Several methods are available to migrate an image mode volume to a managed mode volume:

- If your image mode volume is in the same storage pool as the MDisk you want to migrate to, perform one of these migrations:
  - Migrate a single extent. You must migrate the last extent of the image mode volume (number N-1).
  - Migrate multiple extents.
  - Migrate all of the in-use extents from an MDisk. For example, you must migrate extents off an MDisk that is being deleted.

- If you have two storage pools (one for the image mode volume, and one for the managed mode volumes), you can migrate a volume from one to the other.

Have one storage pool for all the image mode volumes, other storage pools for the managed mode volumes, and use the migrate volume facility.

Be sure to verify that enough extents are available in the target storage pool.

7.4 SAN Volume Controller Migration preparation prerequisites

This section highlights important prerequisites steps to successfully implement the SAN Volume Controller into an existing environment. This process includes the following steps:

- Fabric zoning
- Connect SAN Volume Controller to the fabric for migration
- Remove SAN Volume Controller from the fabric after migration
- Back-End storage consideration
- For the latest support information, see the SAN Volume Controller website at:
- Host attachment

7.4.1 Fabric zoning

This section addresses the basic steps for the following tasks:

- Inserting the SAN Volume Controller and the new storage into a fabric environment
- Making the changes needed when moving the SAN Volume Controller between the storage and the servers
- Removing the SAN Volume Controller from the fabric

These steps assume that the SAN Volume Controller and storage are new to the fabric environment.

Lay out new FC cables to the fabric switch to create the physical connections between the target SAN Volume Controller cluster and the target storage. Have a clear picture of how you want to zone the SAN Volume Controller to the back-end platform. Also, spread the fiber adapter port connections for the most throughput and redundancy. Figure 7-7 on page 201 shows an example with two SAN Volume Controller nodes (one I/O group) and a back-end disk system with eight ports.

The implementation can vary depending on back-end disk systems configuration. Always check vendor recommendations for connectivity.
Chapter 7. SAN Volume Controller-based migration

7.4.2 Connect SAN Volume Controller to the fabric for migration

Start by configuring the zone set for the new target storage unit and the SAN Volume Controller. There are two possibilities for this zoning. Table 7-3 shows the zones created in each case. Zone the target SAN Volume Controller cluster to the target storage, and the source SAN Volume Controller cluster to the source storage.

If the migration uses Metro Mirror, there are a source and target SAN Volume Controller cluster. Two additional zones must be created for the application server and zoned to the source SAN Volume Controller and the target storage.

Table 7-3 Zone requirements for migration

<table>
<thead>
<tr>
<th>Migration using Metro Mirror</th>
<th>Migration using volume migration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source SAN Volume Controller and source storage</td>
<td>SAN Volume Controller and source storage</td>
</tr>
<tr>
<td>Target SAN Volume Controller and target storage</td>
<td>SAN Volume Controller and target storage</td>
</tr>
</tbody>
</table>
If the SAN Volume Controller will remain after migration, additional zones are required for the application server and the target SAN Volume Controller:

- For volume migration, three new zones are required to connect the SAN Volume Controller to both source and target storage. These zones also attach the application server to the new target storage.
- For migration using Metro Mirror, a zone for SAN Volume Controller and the application server are created instead of the application server and target storage.

### 7.4.3 Remove SAN Volume Controller from the fabric after migration

The SAN Volume Controller can remain in the fabric for continued storage virtualization if wanted. To remove the SAN Volume Controller, perform the following steps:

1. Shut down the server.
2. Remove the SAN Volume Controller and application server zone.
3. Create and activate a zone for the application server and the new storage.
4. Remove the SAN Volume Controller zones to the storage.
5. Bring the application server up on the new storage.
6. Disconnect and remove the SAN Volume Controller.

**Remember:** Additional steps on the storage are required to map the application server to the target volumes.

### 7.4.4 Back-End storage consideration

When planning the configuration of external storage systems for use with SAN Volume Controller systems, consider the following details:

- All SAN Volume Controller nodes in a system must be able to connect to the same set of storage system ports on each device. A system that contains nodes that cannot connect to the same set of storage-system ports is considered degraded. In this situation, a system error is logged that requires a repair action. This action is important on storage systems that have exclusion rules that determine to which host bus adapter (HBA) worldwide node names (WWNNs) a storage partition can be mapped.

- A storage-system logical unit (LU) must not be shared between the SAN Volume Controller and a host. You can configure certain storage systems to safely share resources between the SAN Volume Controller system and direct-attached hosts. This type of configuration is described as a split storage system. You must configure the storage system and SAN so the SAN Volume Controller system cannot access LUs that any other host or SAN Volume Controller system can access.
This split storage system configuration can be arranged by storage system LUN mapping and masking. If the split storage system configuration is not guaranteed, data corruption can occur.

The SAN Volume Controller supports configurations where a storage system is split between two SAN Volume Controllers. You must configure the storage system and SAN so that the SAN Volume Controller system cannot access LUs that any other host or SAN Volume Controller system can access. You can use storage system LUN mapping and masking to arrange for this configuration. If this configuration is not guaranteed, data corruption can occur.

**Attention:** Avoid configuring a storage system to present the same LU to more than one SAN Volume Controller system. This configuration is not supported and is likely to cause undetected data loss or corruption.

For the latest support information, see the SAN Volume Controller website at:

http://www-03.ibm.com/systems/storage/software/virtualization/svc/interop.html

### 7.4.5 Unsupported storage systems

When a storage system is detected on the SAN, the SAN Volume Controller attempts to recognize it using its inquiry data. If the device is not supported, the SAN Volume Controller configures the device as a generic device. A generic device might not function correctly when it is addressed by a SAN Volume Controller system, especially under failure scenarios.

However, the SAN Volume Controller system does not regard accessing a generic device as an error condition and does not log an error. Managed disks (MDisks) that are presented by generic devices are not eligible to be used as quorum disks.

### 7.4.6 Host attachment

Host attachment consideration must be taken seriously from compatibility perspective. OS host levels must satisfy latest OS patch levels and SAN compatibility requirements, and be compatible with SAN Volume Controller code level.

The SAN Volume Controller supports IBM and non-IBM hosts. Check the interoperability information at the following web address:

http://www-03.ibm.com/systems/storage/software/virtualization/svc/interop.html

### 7.5 Migrating SAN disks to SAN Volume Controller volumes and back to SAN

This section addresses how to move two LUNs from an AIX server connected to a storage subsystem through SAN over to the SAN Volume Controller.

Manage the LUNs with the SAN Volume Controller. Move them between other managed disks, and then back to image mode disks so that they can be masked and mapped back to the AIX server directly.
Using this procedure allows you to perform any of the following activities in your environment:

- Move the SAN LUNs of an AIX server and virtualize those LUNs through the SAN Volume Controller. Perform these steps when introducing the SAN Volume Controller into your environment.
  
  Your host downtime is only a few minutes while you remap and remask disks using your storage subsystem LUN management tool.

- Move data between storage subsystems while your AIX server is still running and servicing your business application.
  
  Perform this activity if you are removing a storage subsystem from your SAN environment and want to move the data onto more appropriate LUNs. When selecting LUNs, take into account availability, performance, and redundancy.

- Move the LUNs on your AIX server back to image mode volumes so that they can be remapped and remasked directly to the AIX server.

Figure 7-8 shows our AIX server connected to our SAN infrastructure. It has two LUNs (hdisk1 and hdisk2) that are zoned directly to it from our storage subsystem. In this example, a heavy I/O load was simulated to see how the migration process affects the host side.
The `hdisk1` disk makes up the `itsovg01` LVM group, and the `hdisk2` disk makes up the `itsovg02` LVM group, as shown in Example 7-1.

**Example 7-1   AIX SAN configuration**

```
# lspv
hdisk0          00ce0c7b1fcfc0e5                    rootvg          active
hdisk1          00ce0c7b69d8ea93                    itsovg01        active
hdisk2          00ce0c7b69d9454b                    itsovg02        active
#
# lsdev -Cc disk
hdisk0  Available          Virtual SCSI Disk Drive
hdisk1  Available 02-08-02 MPIO 2810 XIV Disk
hdisk2  Available 02-08-02 MPIO 2810 XIV Disk
#
# lscfg |grep -i disk
* hdisk0 U9119.590.83E0C7B-V101-C2-T1-L8100000000000000 Virtual SCSI Disk Drive
* hdisk1 U5791.001.99B0832-P2-C04-T1-W50017380102F0180-L10000000000 MPIO 2810 XIV Disk
* hdisk2 U5791.001.99B0832-P2-C04-T1-W50017380102F0180-L2000000000000000 MPIO 2810 XIV Disk
#
```

The AIX server represents a typical SAN environment. The host directly uses LUNs that were created on a SAN storage subsystem (Figure 7-8 on page 204):

- The HBA cards on the AIX server are zoned so that they are in the Green Zone (dotted line) with the storage subsystem.
- The two LUNs, `hdisk1` and `hdisk2`, are defined on the storage subsystem. They are directly zoned and available to the AIX server.

### 7.5.1 Connecting the SAN Volume Controller to your SAN fabric

This section addresses the steps to take to introduce SAN Volume Controller into your SAN environment. You can accomplish this task without any downtime to any host or application that also uses your storage area network. If you have an SAN Volume Controller already connected, skip to 7.5.2, “Preparing your SAN Volume Controller to virtualize disks” on page 206.

To connect the SAN Volume Controller to your SAN fabric, perform the following tasks:

1. Assemble your SAN Volume Controller components (nodes, UPS, and SSPC Console if available).
2. Cable the SAN Volume Controller correctly.
3. Power the SAN Volume Controller on.
4. Verify that the SAN Volume Controller is visible on your SAN.
5. Create and configure your SAN Volume Controller cluster.
6. Create these additional zones:
   - An SAN Volume Controller node zone (Black Zone)
   - A storage zone (Red Zone)
   - A zone for every host initiator (HBA) (Blue Zone)
7.5.2 Preparing your SAN Volume Controller to virtualize disks

This section describes the preparatory tasks you must perform before taking your AIX server offline. These tasks are all nondisruptive activities and do not affect your SAN fabric or your existing SAN Volume Controller configuration.

Creating a storage pool

When moving the two AIX LUNs to the SAN Volume Controller, they are in image mode, so create a storage pool to hold them. In this example, the storage pool to hold the LUNs is named `aix_imgmdg`.

Create an empty storage pool for these disks using the commands shown in Example 7-2.

Example 7-2 Create empty Storage pool

```
IBM_2145:ITS01:superuser>svctask mkmdiskgrp -name aix_imgmdg -ext 512
MDisk Group, id [2], successfully created
IBM_2145:ITS01:superuser>

IBM_2145:ITS01:superuser>lsmdiskgrp
id name status mdisk_count vdisk_count capacity extent_size free_capacity
virtual_capacity used_capacity real_capacity overallocation warning easy_tier
easy_tier_status
```
Creating host definition

If you prepare the zones correctly, the SAN Volume Controller can see the HBA adapters for the AIX server on the fabric.

First, get the worldwide name (WWN) for the HBA of your AIX server. Make sure that you have the correct WWN to create a zone for every HBA in the Blue Zone and to reduce the AIX server downtime.

Example 7-3 shows the commands to get the WWN. In this example, the host has WWNs of 100000000C95ADE4D and 100000000C9648394.

Example 7-3   Discover your WWN

# lsdev -Cc adapter|grep fcs
fcs0   Available 02-08 FC Adapter
fcs1   Available 03-08 FC Adapter
#
# lscfg -vpl fcs0
fcs0             U5791.001.99B0832-P2-C04-T1  FC Adapter

Part Number.................03N5014
EC Level....................A
Serial Number.............1D719080D1
Manufacturer................001D
Customer Card ID Number....280D
FRU Number..................03N5014
Device Specific.(ZM).......3
Network Address.............100000000C95ADE4D
ROS Level and ID.............02C82138
Device Specific.(Z0).......1036406D
Device Specific.(Z1)......00000000
Device Specific.(Z2)......00000000
Device Specific.(Z3)......03000909
Device Specific.(Z4)......FFC01159
Device Specific.(Z5)......02C82138
Device Specific.(Z6)......06C32138
Device Specific.(Z7)......07C32138
Device Specific.(Z8)......200000000C95ADE4D
Device Specific.(Z9)......BS2.10X8
Device Specific.(ZA)......B1D2.10X8
Device Specific.(ZB)......B2D2.10X8
Device Specific.(ZC)......00000000
Hardware Location Code.....U5791.001.99B0832-P2-C04-T1

PLATFORM SPECIFIC

Name: fibre-channel
Model: LP11000
Node: fibre-channel01
Device Type: fcp
Physical Location: U5791.001.99B0832-P2-C04-T1
# lscfg -vpl fcs1
fcs1 U5791.001.99B082W-P1-C04-T1 FC Adapter

Part Number.................10N8620
Serial Number................1B71704A6C
Manufacturer..................001B
EC Level....................A
Customer Card ID Number.....5759
FRU Number..................10N8620
Device Specific.(ZM).........3
Network Address.............10000000C9648394
ROS Level and ID............02C82138
Device Specific.(Z0)........1036406D
Device Specific.(Z1)........00000000
Device Specific.(Z2)........00000000
Device Specific.(Z3)........03000909
Device Specific.(Z4)........FFC01159
Device Specific.(Z5)........02C82138
Device Specific.(Z6)........06C12138
Device Specific.(Z7)........07C12138
Device Specific.(Z8)........20000000C9648394
Device Specific.(Z9)........8S2.10X8
Device Specific.(ZA)........81F2.10X8
Device Specific.(ZB)........82F2.10X8
Device Specific.(ZC)........00000000
Hardware Location Code......U5791.001.99B082W-P1-C04-T1

PLATFORM SPECIFIC

Name: fibre-channel
Model: LP11000
Node: fibre-channel@1
Device Type: fcp

Physical Location: U5791.001.99B082W-P1-C04-T1

The `svcinfo lshbaportcandidate` command lists all of the WWNs that are not yet allocated to a host on the SAN fabric. Example 7-4 shows the output of the nodes that it found in the example SAN fabric. If the port does not show up, there is a zone configuration problem.

Example 7-4 Add the host to the SAN Volume Controller

```
IBM_2145:ITSO-CLS2:admin>svcinfo lshbaportcandidate
id
10000000C95ADE4D
10000000C9648394
IBM_2145:ITSO-CLS2:admin>
```
After verifying that the SAN Volume Controller can see the host `ITSO_AIX`, create the host entry and assign the WWN to this entry (Example 7-5).

**Example 7-5  Create the host entry**

```
IBM_2145:ITSO-CLS2:admin>svctask mkhost -name ITSO_AIX -hbawwpn 10000000C95ADE4D:10000000C9648394
Host, id [0], successfully created
IBM_2145:ITSO-CLS2:admin>svcinfo lshost ITSO_AIX
id 0
name ITSO_AIX
port_count 2
type generic
mask 1111
iogrp_count 4
WWPN 10000000C9648394
node_logged_in_count 2
state inactive
WWPN 10000000C95ADE4D
node_logged_in_count 2
state inactive
IBM_2145:ITSO1:superuser>
```

**Verifying that you can see your storage subsystem**

Display the storage subsystem in the SAN Volume Controller with the `svcinfo lscontroller` command (Example 7-6). In this example, XIV is used as controller1 with id 1, and will be migrated to DS8000 as a controller with id 0.

**Example 7-6  Discover the storage controller**

```
IBM_2145:ITSO1:superuser>lscontroller
id controller_name ctrl_s/n             vendor_id            product_id_low
product_id_high
0 controller0     75ABTV2FFFF          IBM                  2107900
1 controller1     102F0000             IBM                  2810XIV-
LUN-0
IBM_2145:ITSO1:superuser>
```

**Tip:** The `svctask chcontroller` command allows you to change the discovered storage subsystem name in SAN Volume Controller. In complex SANs, change your storage subsystem to a more meaningful name.

**Getting the disk serial numbers**

To avoid creating the wrong volumes when there are many available unmanaged MDisks that are seen by the SAN Volume Controller, locate the following identifiers:

- The LUN `unique_id` from the AIX side
- The WWN tag from the storage subsystem

In the example, the WWN comes from an XIV disk subsystem.

When you discover these MDisks, confirm that you have the correct serial numbers before creating the image mode volumes.
From the XIV system, use XCLI to obtain the WWN LUN as shown in Example 7-7.

Example 7-7 Use XIV XCLI to obtain WWN for LUN attached to ITSO_AIX host

```
XIV_7804143>>mapping_list host=ITSO_AIX
LUN   Volume   Size   Master   Serial Number   Locked
6     lpar3_1  17              14276           no
7     lpar3_2  17              14277           no
XIV_7804143>>
XIV_7804143>>vol_list vol=lpar3_1 -t "wwn"
WWN
00173800102F37C4
XIV_PFE3_7804143>>vol_list vol=lpar3_2 -t "wwn"
WWN
00173800102F37C5
XIV_PFE3_7804143>>
```

Obtain the hdisk unique_id from ITSO_AIX as shown in Example 7-8.

Example 7-8 Obtain hdisk unique_id from ITSO_AIX

```
# lsattr -El hdisk1 | grep unique_id
unique_id       2611200173800102F37C4072810XIV03IBMfcp

# lsattr -El hdisk2 | grep unique_id
unique_id       2611200173800102F37C5072810XIV03IBMfcp
```

### 7.5.3 Moving the LUNs to the SAN Volume Controller

Next, move the LUNs that are assigned to the AIX server and reassign them to the SAN Volume Controller.

Because you want to move only the LUN that holds your application and data files, move that LUN without rebooting the host. You must unmount only the file system and vary off the volume group (VG) to ensure data integrity after the reassignment.

**Important:** Moving LUNs to the SAN Volume Controller requires that the Subsystem Device Driver is installed on the AIX server. You can install the Subsystem Device Driver ahead of time. However, it might require an outage of your host to do so. The latest driver information is available at following web address:

https://www-304.ibm.com/support/docview.wss?uid=ssg1S7001350

Subsystem Device Driver can coexist with other multipath drivers only during migration.

To move both LUNs at the same time, perform the following steps:

1. Confirm that the Subsystem Device Driver is installed.
2. Unmount the file system and vary off the VGs:
   a. Stop the applications that are using the LUNs.
   b. Unmount those file systems with the `unmount MOUNT_POINT` command.
c. If the file systems are part of a Logical Volume Manager (LVM) volume, deactivate that VG with the `varyoffvg VOLUMEGROUP_NAME` command.

Example 7-9 shows the commands run in the example on ITSO_AIX.

**Example 7-9  AIX command sequence**

```bash
# lslpp -l |grep -i sdd
devices.sddpcm.61.rte       2.6.0.3  COMMITTED IBM SDD PCM for AIX V61
devices.sddpcm.61.rte       2.6.0.3  COMMITTED IBM SDD PCM for AIX V61

# lsfs
Name         Nodename  Mount Pt  VFS  Size Options Auto
Accounting  
/dev/hd4  --         /       jfs2  1048576 --    yes  no
/dev/hd1  --         /home   jfs2  6291456 --    yes  no
/dev/hd2  --         /usr    jfs2  6291456 --    yes  no
/dev/hd9var --         /var    jfs2  3080192 --    yes  no
/dev/hd3  --         /tmp    jfs2  524288 --     yes  no
/dev/hd11admin --        /admin  jfs2  262144 --    yes  no
/proc        --         /proc    procfs --  --     yes  no
/no
/dev/hd10opt --         /opt    jfs2  2097152 --    yes  no
/dev/livedump  --            /var/adm/ras/livedump  jfs2  524288 --    yes  no
/dev/odm   --         /dev/odm  vxodm --       no  no
/dev/fs1v01  --        /itsofs01 jfs2  33423360  rw   yes  no
/dev/fs1v02  --        /itsofs02 jfs2  33423360  rw   yes  no

# lsvg -o
itsovg02
itsovg01
rootvg

# lsvg -o itsovg01
itsovg01:

# lsvg -l itsovg01
itsovg01:
LV NAME   TYPE       LPs  PPs  PVs  LV STATE  MOUNT POINT
loglv01   jfs2log    1  1  1  open/syncd  N/A
fs1v01    jfs2       510 510 1  open/syncd  /itsofs01

# lsvg -l itsovg02
itsovg02:

# lsvg -l itsovg02
itsovg02:
LV NAME   TYPE       LPs  PPs  PVs  LV STATE  MOUNT POINT
loglv02   jfs2log    1  1  1  open/syncd  N/A
fs1v02    jfs2       510 510 1  open/syncd  /itsofs02

# umount /itsofs01
# umount /itsofs02
# varyoffvg itsovg01
# varyoffvg itsovg02

3. From the storage side (the XIV system in the example), unmap the disks from the ITSO_AIX server and remap the disks to the SAN Volume Controller.
4. From the SAN Volume Controller, discover the new disks with the `svctask detectmdisk` command. The disks are discovered and named `mdiskN`, where `N` is the next available `mdisk` number (starting from 0). Example 7-10 shows the commands used in the example to discover the MDisks and verify that you have the correct MDisks. The bolded part shows the LUNs from the XIV system and their unique IDs.

Example 7-10  Discover the new MDisks

```
IBM_2145:ITSO1:superuser>detectmdisk
IBM_2145:ITSO1:superuser>lsmdisk
id name       status         mode      mdisk_grp_id mdisk_grp_name capacity ctrl_LUN_#       controller_name UID tier
8  mdisk8 online unmanaged 16.0GB 0000000000000005 controller1 00173800102f37c4000000000000000000000000000000000000000000000000 generic_hdd
9  mdisk9 online unmanaged 16.0GB 0000000000000006 controller1 00173800102f37c5000000000000000000000000000000000000000000000000 generic_hdd
```

**Important:** Match your discovered MDisk serial numbers (UID in the `svcinfo lsmdisk` command task display) with the serial number that you discovered earlier.

5. After you verify that you have the correct MDisks, rename them to avoid confusion in the future (Example 7-11).

Example 7-11  Rename the MDisks

```
IBM_2145:ITSO1:superuser>svctask chmdisk -name ITSO_AIX01 mdisk8
IBM_2145:ITSO1:superuser>svctask chmdisk -name ITSO_AIX02 mdisk9
IBM_2145:ITSO1:superuser>lsmdisk
id name       status         mode      mdisk_grp_id mdisk_grp_name capacity ctrl_LUN_#       controller_name UID tier
8  ITSO_AIX01 online unmanaged 16.0GB 0000000000000005 controller1 00173800102f37c4000000000000000000000000000000000000000000000000 generic_hdd
9  ITSO_AIX02 online unmanaged 16.0GB 0000000000000006 controller1 00173800102f37c5000000000000000000000000000000000000000000000000 generic_hdd
```

6. Create the image mode volumes with the `svctask mkvdisk` command and the option `-vtype image` (Example 7-12). This command virtualizes the disks in the same layout as though they were not virtualized.

Example 7-12  Create the image mode volumes

```
IBM_2145:ITSO1:superuser>mkvdisk -mdiskgrp aix_imgmdg -iogrp 0 -vtype image -mdisk ITSO_AIX01 -name itso_aix_hdisk1
Virtual Disk, id [2], successfully created

IBM_2145:ITSO1:superuser>mkvdisk -mdiskgrp aix_imgmdg -iogrp 0 -vtype image -mdisk ITSO_AIX02 -name itso_aix_hdisk2
Virtual Disk, id [3], successfully created
```
7. Map the new image mode volumes to the host (Example 7-13).

Example 7-13   Map the volumes to the host

IBM_2145:ITS01:superuser>svctask mkvdiskhostmap -host ITSO_AIX itso_aix_hdisk1
Virtual Disk to Host map, id [2], successfully created

IBM_2145:ITS01:superuser>svctask mkvdiskhostmap -host ITSO_AIX itso_aix_hdisk2
Virtual Disk to Host map, id [3], successfully created

IBM_2145:ITS01:superuser>

Tip: While the application is in a quiescent state, you can use FlashCopy to copy the new image volumes onto other volumes. You do not need to wait until the FlashCopy process is completed before starting your application.

To put the image mode volumes online, perform the following steps:
1. Remove the old disk definitions using the \texttt{rmdev -dl <hdisk>} command.
2. Run the \texttt{cfgmgr -vs} command to rediscover the available LUNs.
3. If your application and data are on an LVM volume, rediscover the VG, and then run the \texttt{varyonvg VOLUME\_GROUP} command to activate the VG.
4. Mount your file systems with the \texttt{mount /MOUNT\_POINT} command.

You are now ready to start your application.

7.5.4 Migrating image mode volumes to volumes

While the AIX server is running and the file systems are in use, migrate the image mode volumes onto striped volumes. The extents are spread over three other MDisks.

Preparing MDisks for striped mode volumes

From the new storage subsystem, perform these tasks:

- Create and allocate three LUNs from your new storage subsystem to the SAN Volume Controller.
- Discover the LUNs as MDisks.
- Rename these LUNs to more meaningful names.
- Create a storage pool.
- Put all these MDisks into this storage pool.

The output of these commands in the example is shown in Example 7-14.

Example 7-14   Creating a storage pool

IBM_2145:ITS01:superuser>mkmdiskgrp -name itso_aix_new -ext 512
MDisk Group, id [1], successfully created
IBM_2145:ITS01:superuser>
IBM_2145:ITS01:superuser>detectmdisk
IBM_2145:ITS01:superuser>
IBM_2145:ITS01:superuser>lsmdisk
id name status mode mdisk_grp_id mdisk_grp_name capacity
ctrl_LUN_# controller_name UID
tier
Migrating the volumes

Migrate the image mode volumes onto striped volumes with the `svctask migratevdisk` command (Example 7-15 on page 215). While the migration is running, the AIX server is still running and you can still access the files.

Before starting the migration process, run heavy I/O load to `hdisk1` and `hdisk2` on the AIX side and collect performance data using the `nmon` tool.

To check the overall progress of the migration, use the `svcinfo lsmigrate` command as shown in Example 7-15 on page 215. Listing the storage pool with the `svcinfo lsmdiskgrp`
command shows that the free capacity on the old storage pool is slowly increasing. Capacity is being freed as those extents are moved to the new storage pool.

Use CLI command \texttt{svctask lsvdiskextent} to track the progress as extents move from image mode volumes to new striped volumes.

\textit{Example 7-15}  Migrating image mode volumes to striped volumes

\begin{verbatim}
IBM_2145:ITSO1:superuser>lsvdiskextent 2
id number_extents
 0  32
IBM_2145:ITSO1:superuser>lsvdiskextent 3
id number_extents
 0  32
IBM_2145:ITSO1:superuser>

IBM_2145:ITSO1:superuser>migratevdisk -vdisk itso_aix_hdisk1 -mdiskgrp itso_aix_new
IBM_2145:ITSO1:superuser>migratevdisk -vdisk itso_aix_hdisk2 -mdiskgrp itso_aix_new

IBM_2145:ITSO1:superuser>lsmigrate
migrate_type MDisk_Group_Migration
 progress 0
migrate_source_vdisk_index 3
migrate_target_mdisk_grp 1
max_thread_count 4
migrate_source_vdisk_copy_id 0
migrate_type MDisk_Group_Migration
 progress 3
migrate_source_vdisk_index 2
migrate_target_mdisk_grp 1
max_thread_count 4
migrate_source_vdisk_copy_id 0

IBM_2145:ITSO1:superuser>lsvdiskextent 2
id number_extents
 5  3
 6  3
 7  3
 8  23

IBM_2145:ITSO1:superuser>lsvdiskextent 3
id number_extents
 5  4
 6  2
 7  3
 9  23

IBM_2145:ITSO1:superuser>lsmigrate
migrate_type MDisk_Group_Migration
 progress 62
migrate_source_vdisk_index 3
migrate_target_mdisk_grp 1
max_thread_count 4
migrate_source_vdisk_copy_id 0
\end{verbatim}
migrate_type MDisk_Group_Migration

progress 65
migrate_source_vdisk_index 2
migrate_target_mdisk_grp 1
max_thread_count 4
migrate_source_vdisk_copy_id 0

IBM_2145:ITSO1:superuser>lsmigrate
IBM_2145:ITSO1:superuser>

IBM_2145:ITSO1:superuser>lsvdiskextent 2
id number_extents
5   11
6   11
7   10
IBM_2145:ITSO1:superuser>lsvdiskextent 3
id number_extents
5   11
6   11
7   10
IBM_2145:ITSO1:superuser>

After this task is completed, Example 7-16 shows that the volumes are spread over three MDisks in the itso_aix_new storage pool. The old storage pool is empty.

Example 7-16   Migration complete

IBM_2145:ITSO1:superuser>svcinfo lsmdiskgrp itso_aix_new
id 1
name itso_aix_new
status online
mdisk_count 3
vdisk_count 2
capacity 43.50GB
extent_size 512
free_capacity 11.50GB
virtual_capacity 32.00GB
used_capacity 32.00GB
real_capacity 32.00GB
overallocation 73
warning 0
easy_tier auto
easy_tier_status inactive
tier generic_ssd
tier_mdisk_count 0
tier_capacity 0.00MB
tier_free_capacity 0.00MB
tier generic_hdd
tier_mdisk_count 3
tier_capacity 43.50GB
tier_free_capacity 11.50GB
IBM_2145:ITSO1:superuser>

IBM_2145:ITSO1:superuser>svcinfo lsmdiskgrp aix_imgmdg
id 2
name aix_imgmdg
The migration to the SAN Volume Controller is complete. You can remove the original MDisks from the SAN Volume Controller, and the LUNs from the storage subsystem.

If the LUNs are the ones used last on your storage subsystem, remove them from your SAN fabric using the following steps, shown in Example 7-17:

1. Remove MDisks from the storage pool.
2. Remove the storage pool.
3. Delete the storage pool.
4. Remove mapping from SAN Volume Controller on your back-end storage.
5. Run the `detectmdisk` command on SAN Volume Controller to be sure that MDisks are gone.

**Example 7-17  Using SAN Volume Controller to remove the storage pool**

```
IBM_2145:ITS01:superuser>lsmdiskgrp
id name status mdisk_count vdisk_count capacity extent_size free_capacity
virtual_capacity used_capacity real_capacity overallocation warning easy_tier
easy_tier_status
```

IBM_2145:ITSO1:superuser> lsmdisk
id name         status         mode      mdisk_grp_id mdisk_grp_name capacity ctrl_LUN_#       controller_name UID tier
5  itso_aix_md1 online         managed   1            itso_aix_new   15.0GB
6  itso_aix_md2 online         managed   1            itso_aix_new   15.0GB
7  itso_aix_md3 online         managed   1            itso_aix_new   15.0GB
8  ITSO_AIX01 online managed 2            aix_imgmdg     16.0GB
9  ITSO_AIX02 online managed 2            aix_imgmdg     16.0GB

IBM_2145:ITSO1:superuser> rmmdisk -mdisk ITSO_AIX01 aix_imgmdg
IBM_2145:ITSO1:superuser> rmmdisk -mdisk ITSO_AIX02 aix_imgmdg

IBM_2145:ITSO1:superuser> lsmdiskgrp
id name         status mdisk_count vdisk_count capacity extent_size free_capacity
virtual_capacity used_capacity real_capacity overallocation warning easy_tier
easy_tier_status
1  itso_aix_new online 3           2           43.50GB  512         11.50GB
32.00GB          32.00GB       32.00GB       73             0       auto
inactive
2  aix_imgmdg online 0           0           0        512         0
0.00MB           0.00MB        0.00MB        0              0       auto
inactive

IBM_2145:ITSO1:superuser> lsmdisk
id name         status         mode      mdisk_grp_id mdisk_grp_name capacity ctrl_LUN_#       controller_name UID tier
5  itso_aix_md1 online         managed   1            itso_aix_new   15.0GB
6  itso_aix_md2 online         managed   1            itso_aix_new   15.0GB
7  itso_aix_md3 online         managed   1            itso_aix_new   15.0GB
8  ITSO_AIX01 online managed 2            aix_imgmdg     16.0GB
0000000000000000 controller1
00173800102f37c3000000000000000000000000000000000000000000000000 generic_hdd
9  ITSO_AIX02 online managed 2            aix_imgmdg     16.0GB
0000000000000000 controller1
00173800102f37c3000000000000000000000000000000000000000000000000 generic_hdd
Chapter 7. SAN Volume Controller-based migration

7.5.5 Performance analysis

As shown in Figure 7-10 on page 220, there was no performance impact to host I/O during the test. During the migration, there was higher oscillation in the I/O profile, but SAN Volume Controller sustained I/O requests from the host side.

On the graph shows three segments:

- Before migration
- During migration
- Post migration

With the I/O pattern constant during the test, I/O activity increased after moving from image mode MDisk (not striped) to stripe MDisk. Therefore, always use striping to improve performance.

I/O Pattern: BS=4k, Mixed R/W 50/50, random, 50% Cache Hit
7.5.6 Preparing to migrate from the SAN Volume Controller

Before changing the AIX server LUNs from being accessed by the SAN Volume Controller as volumes to being directly accessed, convert the volumes into image mode volumes.

This activity might be needed for one of these reasons:

- You purchased a new storage subsystem, and are using the SAN Volume Controller as a tool to migrate the new storage system.
- You used the SAN Volume Controller to FlashCopy or Metro Mirror a volume onto another volume, and no longer need that host connected to the controller.
- You want to move a host and its data currently connected to the SAN Volume Controller to a site where there is no SAN Volume Controller.
- Changes to your environment no longer require this host to use the SAN Volume Controller.

There are other preparatory activities to be performed before we shut down the host and reconfigure the zoning and LUN mapping. This section covers those activities.
If you are moving the data to a new storage subsystem, the subsystem must be connected to your SAN fabric, powered on, and visible from your SAN switches. Your environment should look similar to the environment shown in Figure 7-12.

To prepare for the migration, perform the following steps:

- Set up the SAN configuration so that all of the zones are created.
- Add the new storage subsystem to the Red Zone so that the SAN Volume Controller can communicate with it directly.
- Create a Green Zone for the host to use when you are ready for it to directly access the disk. This step must take place after the host is removed from the SAN Volume Controller.

After your zone configuration is set up correctly, use the `svcinfo lscontroller` command to force SAN Volume Controller to recognize the new storage subsystem controller (Example 7-18). In addition, you might want to rename the controller to a more meaningful name using the `svctask chcontroller -name` command.

**Example 7-18  Discovering the new storage subsystem**

```
IBM_2145:ITS01:superuser>lscontroller
id controller_name ctrl_s/n vendor_id product_id_low
product_id_high
0 controller0 75ABTV2FFFF IBM 2107900
1 controller1 102F0000 IBM 2810XIV-LUN-0
IBM_2145:ITS01:superuser>
```
7.5.7 Creating new LUNs

The example storage subsystem has two LUNs that are masked so the SAN Volume Controller can see them. These LUNs link directly to the host, removing the volumes that it currently has. To check that the SAN Volume Controller can use the LUNs, issue the svctask detectmdisk command as shown in Example 7-19.

In the example, two 16 GB LUNs are on the XIV subsystem. Migrate them back to image mode volumes and the XIV subsystem in one step.

Example 7-19 Discover the new MDisks

```
 IBM_2145:ITSO1:superuser> detectmdisk
 IBM_2145:ITSO1:superuser> lsmdisk

 5 itso_aix_md1 online managed 1 itso_aix_new 15.0GB
 4084400500000000 controller0
6005076303ffe630000000000000840500000000000000000000000000000000 generic_hdd

 6 itso_aix_md2 online managed 1 itso_aix_new 15.0GB
 4085400400000000 controller0
6005076303ffe630000000000000850400000000000000000000000000000000 generic_hdd

 7 itso_aix_md3 online managed 1 itso_aix_new 15.0GB
 4085400500000000 controller0
6005076303ffe630000000000000850500000000000000000000000000000000 generic_hdd

 8 mdisk5 online unmanaged 16.0GB
 0000000000000005 controller1
00173800102f37c4000000000000000000000000000000000000000000000000 generic_hdd

 9 mdisk6 online unmanaged 16.0GB
 0000000000000006 controller1
00173800102f37c5000000000000000000000000000000000000000000000000 generic_hdd
```

Although the MDisks do not stay in the SAN Volume Controller for long, rename them. Change the names to more meaningful ones so that you do not confuse them with other MDisks. Also, create the storage pools to hold your new MDisks as shown in Example 7-20.

Example 7-20 Rename the MDisks

```
 IBM_2145:ITSO1:superuser> svctask chmdisk -name AIX_MIG01 mdisk5
 IBM_2145:ITSO1:superuser> svctask chmdisk -name AIX_MIG02 mdisk6
 IBM_2145:ITSO1:superuser> svctask mkmdiskgrp -name ITSO_AIXMIG -ext 512
 MDisk Group, id [2], successfully created

 IBM_2145:ITSO1:superuser> lsmdiskgrp

 id name status mdisk_count capacity extent_size free_capacity virtual_capacity used_capacity real_capacity overallocation warning easy_tier easy_tier_status
 1 itso_aix_new online 3 2 43.50GB 512 11.50GB 32.00GB 32.00GB 32.00GB 73 0 auto inactive
 2 ITSO_AIXMIG online 0 0 0 512 0 0.00MB 0.00MB 0.00MB 0 0 auto inactive

 IBM_2145:ITSO1:superuser>
```

Your SAN Volume Controller environment is now ready for the volume migration to image mode volumes.
7.5.8 Migrating the managed volumes
While the AIX server is still running, migrate the managed volumes onto the new MDisks
using image mode volumes. Perform this action with the svctask migratetoimage command
as shown in Example 7-21.
Example 7-21 Migrate the volumes to image mode volumes

IBM_2145:ITSO1:superuser>migratetoimage -vdisk itso_aix_hdisk1 -mdisk AIX_MIG01
-mdiskgrp ITSO_AIXMIG
IBM_2145:ITSO1:superuser>migratetoimage -vdisk itso_aix_hdisk2 -mdisk AIX_MIG02
-mdiskgrp ITSO_AIXMIG
IBM_2145:ITSO1:superuser>lsmdisk
id name
status
mode
mdisk_grp_id mdisk_grp_name capacity
ctrl_LUN_#
controller_name UID
tier
5 itso_aix_md1 online
managed
1
itso_aix_new
15.0GB
4084400500000000 controller0
6005076303ffce63000000000000840500000000000000000000000000000000 generic_hdd
6 itso_aix_md2 online
managed
1
itso_aix_new
15.0GB
4085400400000000 controller0
6005076303ffce63000000000000850400000000000000000000000000000000 generic_hdd
7 itso_aix_md3 online
managed
1
itso_aix_new
15.0GB
4085400500000000 controller0
6005076303ffce63000000000000850500000000000000000000000000000000 generic_hdd
8 AIX_MIG01
online
image
2
ITSO_AIXMIG
16.0GB
0000000000000005 controller1
00173800102f37c4000000000000000000000000000000000000000000000000 generic_hdd
9 AIX_MIG02
online
image
2
ITSO_AIXMIG
16.0GB
0000000000000006 controller1
00173800102f37c5000000000000000000000000000000000000000000000000 generic_hdd
IBM_2145:ITSO1:superuser>lsmigrate
migrate_type Migrate_to_Image
progress 25
migrate_source_vdisk_index 2
migrate_target_mdisk_index 8
migrate_target_mdisk_grp 2
max_thread_count 4
migrate_source_vdisk_copy_id 0
migrate_type Migrate_to_Image
progress 3
migrate_source_vdisk_index 3
migrate_target_mdisk_index 9
migrate_target_mdisk_grp 2
max_thread_count 4
migrate_source_vdisk_copy_id 0
IBM_2145:ITSO1:superuser>
IBM_2145:ITSO1:superuser>lsvdiskextent 2
id number_extents
5 4
6 5
7 3
8 20

Chapter 7. SAN Volume Controller-based migration

223


IBM_2145:ITSO1:superuser>lsvdiskextent 3
id number_extents
5 6
6 7
7 6
9 13
IBM_2145:ITSO1:superuser>lsmigrate
IBM_2145:ITSO1:superuser>
IBM_2145:ITSO1:superuser>lsvdiskextent 2
id number_extents
8 32
IBM_2145:ITSO1:superuser>lsvdiskextent 3
id number_extents
9 32
IBM_2145:ITSO1:superuser>IBM_2145:ITSO1:superuser>migratetoimage -vdisk
itso_aix_hdisk1 -mdisk AIX_MIG01 -mdiskgrp ITSO_AIXMIG
IBM_2145:ITSO1:superuser>migratetoimage -vdisk itso_aix_hdisk2 -mdisk AIX_MIG02
-mdiskgrp ITSO_AIXMIG
IBM_2145:ITSO1:superuser>lsmdisk
id name
status
mode
mdisk_grp_id mdisk_grp_name capacity
ctrl_LUN_#
controller_name UID
tier
5 itso_aix_md1 online
managed
1
itso_aix_new
15.0GB
4084400500000000 controller0
6005076303ffce63000000000000840500000000000000000000000000000000 generic_hdd
6 itso_aix_md2 online
managed
1
itso_aix_new
15.0GB
4085400400000000 controller0
6005076303ffce63000000000000850400000000000000000000000000000000 generic_hdd
7 itso_aix_md3 online
managed
1
itso_aix_new
15.0GB
4085400500000000 controller0
6005076303ffce63000000000000850500000000000000000000000000000000 generic_hdd
8 AIX_MIG01
online
image
2
ITSO_AIXMIG
16.0GB
0000000000000005 controller1
00173800102f37c4000000000000000000000000000000000000000000000000 generic_hdd
9 AIX_MIG02
online
image
2
ITSO_AIXMIG
16.0GB
0000000000000006 controller1
00173800102f37c5000000000000000000000000000000000000000000000000 generic_hdd
IBM_2145:ITSO1:superuser>lsmigrate
migrate_type Migrate_to_Image
progress 25
migrate_source_vdisk_index 2
migrate_target_mdisk_index 8
migrate_target_mdisk_grp 2
max_thread_count 4
migrate_source_vdisk_copy_id 0
migrate_type Migrate_to_Image
progress 3
migrate_source_vdisk_index 3
migrate_target_mdisk_index 9
migrate_target_mdisk_grp 2

224

Data Migration to IBM Disk Storage Systems


During the migration, the AIX server is unaware that its data is being physically moved between storage subsystems.

After the migration is complete, the image mode volumes are ready to be removed from the AIX server. The real LUNs can be mapped and masked directly to the host by using the storage subsystems tool.

### 7.5.9 Removing the LUNs from the SAN Volume Controller

The next step requires a downtime while you remap and remask the disks so the host sees them directly through the Green Zone. Because your LUNs hold only data files and you are using a unique VG, you can remap and remask the disks without rebooting the host. The only requirement is that you unmount the file system and vary off the VG to ensure data integrity after the reassignment.

**Remember:** Moving LUNs to another storage system might need a driver other than SDD. Check with the storage subsystems vendor to see which driver you need. You might be able to install this driver ahead of time.

Follow these steps to remove the SAN Volume Controller:

1. Confirm that the correct device driver for the new storage subsystem is loaded. In the example, you have already installed the XIV host connection kit for AIX.
2. Shut down any applications and unmount the file systems:
   a. Stop the applications that are using the LUNs.
b. Unmount those file systems with the `umount MOUNT_POINT` command.

c. If the file systems are an LVM volume, deactivate that VG with the `varyoffvg VOLUMEGROUP_NAME` command.

3. Remove the volumes from the host by using the `svctask rmvdiskhostmap` command (Example 7-22). To double-check that the volumes are removed, use the `svcinfo lshostvdiskmap` command. The command shows that these disks are no longer mapped to the AIX server.

   **Example 7-22  Remove the volumes from the host**
   
   IBM_2145:ITSO1:superuser>rmvdiskhostmap -host ITSO_AIX itso_aix_hdisk1
   IBM_2145:ITSO1:superuser>rmvdiskhostmap -host ITSO_AIX itso_aix_hdisk2
   IBM_2145:ITSO1:superuser>lshostvdiskmap ITSO_AIX
   IBM_2145:ITSO1:superuser>

4. Remove the volumes from the SAN Volume Controller using the `svctask rmvdisk` command, which makes the MDisks unmanaged as shown in Example 7-23.

   **Cached data:** When you run the `svctask rmvdisk` command, the SAN Volume Controller first double-checks that there is no outstanding dirty cached data for the volume being removed. If uncommitted cached data still exists, the command fails with the following error message:

   CMMVC6212E The command failed because data in the cache has not been committed to disk

   You must wait for this cached data to be committed to the underlying storage subsystem before you can remove the volume.

   The SAN Volume Controller will automatically destage uncommitted cached data two minutes after the last write activity for the volume. How much data there is to destage, and how busy the I/O subsystem is, determine how long this command takes to complete.

   You can check whether the volume has uncommitted data in the cache by using the `svcinfo lsvdisk <VDISKNAME>` command and checking the `fast_write_state` attribute. This attribute has the following meanings:

   - **empty**  No modified data exists in the cache.
   - **not_empty**  Modified data might exist in the cache.
   - **corrupt**  Modified data might have existed in the cache, but any modified data has been lost.

   **Example 7-23  Removing volumes from the SAN Volume Controller**
   
   IBM_2145:ITSO1:superuser>rmvdisk itso_aix_hdisk1
   IBM_2145:ITSO1:superuser>rmvdisk itso_aix_hdisk2
   IBM_2145:ITSO1:superuser>lsvdisk
   id name         status         mode      mdisk_grp_id mdisk_grp_name capacity ctrl_LUN_#  controller_name UID tier
   5 itso_aix_md1 online         managed   1            itso_aix_new   15.0GB 40844005000000000 controller0
   6 itso_aix_md2 online         managed   1            itso_aix_new   15.0GB 40854004000000000 controller0
5. Unmap LUNs from the AIX, and zone the disks from the SAN Volume Controller back to the AIX server.

<table>
<thead>
<tr>
<th>LUN Name</th>
<th>Status</th>
<th>Command</th>
<th>Size</th>
<th>Controller ID</th>
<th>Disk Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>itso_aix_md3</td>
<td>online</td>
<td>cfgmgr -S</td>
<td>15.0GB</td>
<td>controller0</td>
<td>generic_hdd</td>
</tr>
<tr>
<td>AIX_MIG01</td>
<td>online</td>
<td>lsdev -Cc disk</td>
<td>16.0GB</td>
<td>controller1</td>
<td>generic_hdd</td>
</tr>
<tr>
<td>AIX_MIG02</td>
<td>online</td>
<td>lsdev -Cc disk</td>
<td>16.0GB</td>
<td>controller1</td>
<td>generic_hdd</td>
</tr>
</tbody>
</table>

**Important:** This step is the last step that you can perform and still safely back out of everything you have done so far. Up to this point, you can reverse all of the actions that you have performed so far to get the server back online without data loss. However, after you start the next step, you might not be able to turn back without data loss.

To access the LUNs from the AIX server, perform the following steps:

1. Run the `cfgmgr -S` command to discover the storage subsystem.
2. Use the `lsdev -Cc disk` command to verify the discovery of the new disk.
3. Remove the references to all of the old disks.
4. If your application and data are on an LVM volume, rediscover the VG and then run the `varyonvg VOLUME_GROUP` command to activate the VG.
5. Mount your file systems with the `mount /MOUNT_POINT` command.

To make sure that the MDisks are removed from the SAN Volume Controller, run the `svctask detectmdisk` command. The MDisks are first discovered as offline. Then they will automatically be removed after the SAN Volume Controller determines that there are no volumes associated with them.

### 7.6 SAN Volume Controller Volume migration between two storage pools

This section addresses the specific steps to perform volume migration for AIX using SAN Volume Controller between storage pools. If you already have SAN Volume Controller implemented in environment, using SAN Volume Controller Volume migration functionality is a non-disruptive way to migrate data between storage pools. For more information about this process, see 7.3.3, “Migrating a volume between storage pools” on page 192.

If you are implementing SAN Volume Controller for the first time, see 7.2, “SAN Volume Controller concepts for migrating the data” on page 185.

Using this feature, you can easily migrate the data between two physically different storage subsystems, fully not apparent to the application.
7.6.1 Environment description

The environment used for this migration example is shown in Figure 7-13. In this environment, SAN Volume Controller is already in place with back-end storage. New back-end storage will be added that will be used to migrate the data.

In the example, the AIX system is named ITSO_AIX and the SAN Volume Controller cluster is named ITSO1. There are two storage zones in use during the migration.

When using the SAN Volume Controller Volume migration feature for data migration, perform the following basic steps:

1. Create the storage pool from LUNs created on target storage system.
2. Migrate the volumes between the storage pool that holds them to the target storage pool.
3. Check that data is consistent after the migration is finished.
4. Delete the source storage pool if not use in a future. This step is optional.

**Important:** Volume mirroring method can be used to minimize the impact to the volume because the volume goes offline only if the source storage pool fails.

Figure 7-13 shows the environment used during testing. As described previously, from the host side this process is a not apparent and non-disruptive way to migrate data.

The example uses the following components and tools:

1. AIX 6.1 Server as application host
2. Latest IBM SDD drivers installed
3. IBM SAN Volume Controller, with code release 6.2.0.0
4. Two backend disk subsystems
5. The `dd` command to simulate read/write I/O operations to file system during migration
6. The `nmon` tool to collect disk I/O activity for performance data

**Tip:** Keep your OS level, HBA adapter, SAN environment, SAN Volume Controller code level, and back-end storage systems up to date to minimize any potential problems.

Table 7-4 shows the AIX host configuration for this test.

<table>
<thead>
<tr>
<th>System Model</th>
<th>IBM,9119-590</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor type</td>
<td>POWER5</td>
</tr>
<tr>
<td>Number of processors</td>
<td>1</td>
</tr>
<tr>
<td>Processor clock speed</td>
<td>1654 MHz</td>
</tr>
<tr>
<td>CPU type</td>
<td>64 bit</td>
</tr>
<tr>
<td>Memory size</td>
<td>1024</td>
</tr>
</tbody>
</table>

Before starting the migration, verify the server information as shown in Example 7-24. The example uses a single file system built in a volume group. It consists of two LUNs (`hdisk41` and `hdisk42`).

**Example 7-24**  AIX environment before migration

```
# lsdev -Cc disk
hdisk0  Available          Virtual SCSI Disk Drive
hdisk41 Available 02-08-02 MPIO FC 2145
hdisk42 Available 02-08-02 MPIO FC 2145

# datapath query device
DEV#:  41  DEVICE NAME:   hdisk41  TYPE: 2145  ALGORITHM:  Load Balance
SERIAL: 60050768018106209800000000000007
---------------------------------------------------------------------------
Path#       Adapter/Path Name          State     Mode     Select     Errors
     0*          fscsi0/path0           OPEN   NORMAL         56          0
     1*          fscsi0/path1           OPEN   NORMAL         56          0
     2           fscsi0/path2           OPEN   NORMAL      30994          0
     3           fscsi0/path3           OPEN   NORMAL      31071          0
     4*          fscsi1/path4           OPEN   NORMAL         56          0
     5*          fscsi1/path5           OPEN   NORMAL         56          0
     6           fscsi1/path6           OPEN   NORMAL      29884          0
     7           fscsi1/path7           OPEN   NORMAL      29797          0

DEV#:  42  DEVICE NAME:   hdisk42  TYPE: 2145  ALGORITHM:  Load Balance
SERIAL: 60050768018106209800000000000008
---------------------------------------------------------------------------
Path#       Adapter/Path Name          State     Mode     Select     Errors
     0          fscsi0/path0           OPEN   NORMAL  112087          0
     1          fscsi0/path1           OPEN   NORMAL  112497          0
     2*          fscsi0/path2           OPEN   NORMAL         56          0
```
Verify the SAN Volume Controller information as shown in Example 7-25.

Example 7-25  SAN Volume Controller environment

IBM_2145:ITSO1:superuser>lscluster
id  name  location  partnership  bandwidth  id_alias
0000000000000001  ITSO1  local
IBM_2145:ITSO1:superuser>

IBM_2145:ITSO1:superuser>lsmdisk
id name  status  mode  mdisk_grp_id  mdisk_grp_name  capacity  ctrl_LUN_#  controller_name  UID
0  mdisk1  online  managed  1  ITSO_old  16.0GB  00173800102f38d60000000000000000  generic_hdd
1  mdisk0  online  managed  1  ITSO_old  16.0GB  00173800102f38d70000000000000000  generic_hdd
2  mdisk3  online  managed  1  ITSO_old  16.0GB  00173800102f38d80000000000000000  generic_hdd
3  mdisk2  online  managed  1  ITSO_old  16.0GB  00173800102f38d90000000000000000  generic_hdd
4  mdisk4  degraded_ports  unmanaged  15.0GB  4084400400000000  controller0
5  mdisk5  online  managed  0  ITSO_new  15.0GB  4084400500000000  controller0
6  mdisk6  online  managed  0  ITSO_new  15.0GB  4085400400000000  controller0
7  mdisk7  online  managed  0  ITSO_new  15.0GB  4085400500000000  controller0
IBM_2145:ITSO1:superuser>

IBM_2145:ITSO1:superuser>lsmdiskgrp
On the SAN Volume Controller side, the storage pools and volumes are already created. For more information about how to create storage pools and volumes, and how to map volumes to the hosts, see 7.5, “Migrating SAN disks to SAN Volume Controller volumes and back to SAN” on page 203 or see Implementing the IBM System Storage SAN Volume Controller V6.1, SG24-7933.
7.6.2 Performance measurement

To show the performance behavior from the host side, use the `dd` command to read/write I/O from the `/svc_itso` file system. Also, run `noon` in parallel to collect performance data to show the performance behavior and any performance impact during the migration phase.

7.6.3 Migration steps

To migrate the data, perform the following steps:

1. Start I/O load to the `/svc_itso` file system. Example 7-26 shows I/O load from the host side.

   `Example 7-26  AIX I/O load`

<table>
<thead>
<tr>
<th>Disk-Adapter-I/O</th>
<th>Name</th>
<th>%busy</th>
<th>read</th>
<th>write</th>
<th>xfers</th>
<th>Disks</th>
<th>Adapter-Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>fcs0</td>
<td>63.1</td>
<td>5287.6</td>
<td>5501.8</td>
<td>KB/s</td>
<td>2697.3</td>
<td>24 FC Adapter</td>
</tr>
<tr>
<td></td>
<td>fcs1</td>
<td>73.1</td>
<td>5125.4</td>
<td>5205.5</td>
<td>KB/s</td>
<td>2582.2</td>
<td>24 FC Adapter</td>
</tr>
<tr>
<td></td>
<td>vscsi0</td>
<td>1.0</td>
<td>0.0</td>
<td>10.0</td>
<td>KB/s</td>
<td>2.5</td>
<td>1 Virtual SCSI Client A</td>
</tr>
<tr>
<td>TOTALS</td>
<td>3 adapters</td>
<td>10413.0</td>
<td>10717.3</td>
<td>KB/s</td>
<td>5282.1</td>
<td>49 TOTAL(Mbps)=20.6</td>
<td></td>
</tr>
</tbody>
</table>

   Disk-KBytes/second-(K=1024,M=1024*1024)

<table>
<thead>
<tr>
<th>Disk</th>
<th>Busy</th>
<th>Read</th>
<th>Write</th>
<th>Transfers</th>
<th>Size</th>
<th>Peak%</th>
<th>Peak KB/s qDepth</th>
</tr>
</thead>
<tbody>
<tr>
<td>hdisk41</td>
<td>71%</td>
<td>4796.1</td>
<td>4902.1</td>
<td>2424.6</td>
<td>4.0</td>
<td>75%</td>
<td>12330.5</td>
</tr>
<tr>
<td>hdisk42</td>
<td>73%</td>
<td>5160.1</td>
<td>5380.1</td>
<td>2635.1</td>
<td>4.0</td>
<td>83%</td>
<td>10949.9</td>
</tr>
<tr>
<td>Totals(MBps)</td>
<td>Read=9.7</td>
<td>Write=10.0</td>
<td>Size(GB)=3983</td>
<td>Free(GB)=18</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

   Figure 7-14 shows I/O load from SAN Volume Controller side.

2. Start the migration process for volumes AIX_hdisk41 and AIX_hdisk42 using the `migratevdisk` command as shown in Example 7-27. This process moves the volumes from the ITSO_old storage pool to the ITSO_new storage pool.

   `Example 7-27  SAN Volume Controller migratevdisk command`

   `IBM_2145:ITSO1:superuser>svctask migratevdisk -copy 0 -mdiskgrp ITSO_new -vdisk AIX_hdisk41`
3. During the migration process, track the progress using the `lsmigrate` command. You can also track the extent level migration with the `lsvdiskextent` command as shown in Example 7-28.

```
Example 7-28  SAN Volume Controller migration progress

IBM_2145:ITSO1:superuser>lsmigrate
migrate_type MDisk_Group_Migration
progress 10
migrate_source_vdisk_index 1
migrate_target_mdisk_grp 0
max_thread_count 4
migrate_source_vdisk_copy_id 0
migrate_type MDisk_Group_Migration
progress 30
migrate_source_vdisk_index 0
migrate_target_mdisk_grp 0
max_thread_count 4
migrate_source_vdisk_copy_id 0
IBM_2145:ITSO1:superuser>

IBM_2145:ITSO1:superuser>lsvdiskextent 0
id number_extents
 0 5
 1 5
 2 6
 3 5
 5 7
 6 6
 7 6
IBM_2145:ITSO1:superuser>

IBM_2145:ITSO1:superuser>lsvdiskextent 1
id number_extents
 0 12
 1 11
 2 10
 3 11
 5 6
 6 5
 7 5
IBM_2145:ITSO1:superuser>
```

**Tip:** The migration process can take some time. The amount depends on the size of your volumes and the performance of your source and target storage systems. Calculate the time needed before you start the migration so you have good estimations for your environment.

Your IBM technical representative can do a performance assessment before you go live with your migration.
4. When the migration is finished, check the migration status with the `lsmigrate` and `lsvdiskextent` commands as shown in Example 7-29. In the example, `lsmigrate` shows that there is no running migration process. The `lsvdiskextent` command lists the MDisk extents for the volumes. There are the same number of extents as there were before starting the migration process.

```
Example 7-29   SAN Volume Controller migration status

IBM_2145:ITS01:superuser>lsmigrate
IBM_2145:ITS01:superuser>

IBM_2145:ITS01:superuser>lsvdiskextent 0
id number_extents
5  14
6  13
7  13

IBM_2145:ITS01:superuser>lsvdiskextent 1
id number_extents
5  20
6  20
7  20

IBM_2145:ITS01:superuser>
```

Volumes AIX_hdisk41 and AIX_hdisk42 are moved to new storage pool ITSO_new as shown in Example 7-30.

```
Example 7-30   SAN Volume Controller volumes status

IBM_2145:ITS01:superuser>lsvdisk
id name IO_group_id IO_group_name status mdisk_grp_id mdisk_grp_name capacity
type FC_id FC_name RC_id RC_name vdisk_UID fc_map_count copy_count
fast_write_state se_copy_count
0 AIX_hdisk41 0 io_grp0 online 0 ITSO_new 10.00GB striped
60050768018106209800000000000007 0 1 empty 0

1 AIX_hdisk42 0 io_grp0 online 0 ITSO_new 15.00GB striped
60050768018106209800000000000008 0 1 not_empty 0

IBM_2145:ITS01:superuser>
```

5. Delete storage pool ITSO_old and all assigned Mdisks (Example 7-31).

```
Example 7-31   SAN Volume Controller removing storage pool

IBM_2145:ITS01:superuser>rmmdiskgrp -force ITSO_old
IBM_2145:ITS01:superuser>
```

**Attention:** Always be careful when using the `-force` flag because it forces action without any warning.
6. From the AIX side, check the volume, physical volumes, and file systems. Everything should remain the same after migration is complete and the storage pool ITSO_old is deleted as shown in Example 7-32.

**Example 7-32  AIX after migration is finished**

```
# lsdev -Cc disk
hdisk0  Available  Virtual SCSI Disk Drive
hdisk41 Available 02-08-02 MPIO FC 2145
hdisk42 Available 02-08-02 MPIO FC 2145

# datapath query device
DEV#: 41  DEVICE NAME: hdisk41  TYPE: 2145  ALGORITHM: Load Balance
SERIAL: 60050768018106209800000000000007
Path#:  Adapter/Path Name       State     Mode     Select     Errors
0*     fscsi0/path0           OPEN   NORMAL         56          0
1*     fscsi0/path1           OPEN   NORMAL         56          0
2     fscsi0/path2           OPEN   NORMAL    1628458          0
3     fscsi0/path3           OPEN   NORMAL    1626008          0
4*     fscsi1/path4           OPEN   NORMAL         56          0
5*     fscsi1/path5           OPEN   NORMAL         56          0
6     fscsi1/path6           OPEN   NORMAL    1544459          0
7     fscsi1/path7           OPEN   NORMAL    1544973          0
DEV#: 42  DEVICE NAME: hdisk42  TYPE: 2145  ALGORITHM: Load Balance
SERIAL: 60050768018106209800000000000008
Path#:  Adapter/Path Name       State     Mode     Select     Errors
0     fscsi0/path0           OPEN   NORMAL    1562227          0
1     fscsi0/path1           OPEN   NORMAL    1562209          0
2*     fscsi0/path2           OPEN   NORMAL         56          0
3*     fscsi0/path3           OPEN   NORMAL         56          0
4     fscsi1/path4           OPEN   NORMAL    1527249          0
5     fscsi1/path5           OPEN   NORMAL    1526364          0
6*     fscsi1/path6           OPEN   NORMAL         56          0
7*     fscsi1/path7           OPEN   NORMAL         56          0

# lspv
hdisk0          00ce0c7b1fcfc0e5                    rootvg          active
hdisk41         00ce0c7b4b252b09                    svc_itso_vg     active
hdisk42         00ce0c7b4b252c20                    svc_itso_vg     active

# df -k /svc_itso
file system    1024-blocks      Free %Used    Iused %Iused Mounted on
/dev/fslv00      26116096  26110500    1%        6     1% /svc_itso
```

7.6.4 Performance Analyses

This section highlights the performance impact during migration in the example. The graphs are divided into the following sections:

- Old Storage pool (ITSO_old) before migration
- Migration
- New Storage pool (ITSO_new) after migration
You can also see where the I/O load was active and when it was stopped.

Figure 7-15 shows overall System Summary for ITSO_AIX during the example migration.

![Figure 7-15 SAN Volume Controller performance overview: ITSO_AIX System summary](image)

Figure 7-16 shows the Disk Busy graph for ITSO_AIX.

![Figure 7-16 SAN Volume Controller performance overview: ITSO_AIX Host Disk Busy](image)

Migration using SAN Volume Controller does affect the host I/O performance in the example environment, but only for a short time. This performance impact is closely connected to the back-end disk subsystem and its performance profile.

The test used the default number of threads (four) on SAN Volume Controller. Four threads is the maximum value and it can saturate back-end disk system, as you can see in the example. However, the process can be prioritized by specifying the number of threads to be used in parallel (from 1 to 4) while migrating. Using a single thread creates the least background load on the system.

You can also see in the graphs that the data was migrated from a higher performance to a lower performance tier storage.
7.7 Data migration using SAN Volume Controller mirrored volumes

This section shows how to use the volume mirroring feature in SAN Volume Controller to migrate data from one storage device to another. If you already have SAN Volume Controller implemented in your environment, using the SAN Volume Controller mirroring volumes function is a non-disruptive way to migrate data.

Using this feature, you can easily migrate data between two physically separate storage subsystems. The process is fully not apparent to the application. Volume mirroring is included in the base virtualization license, so you do not need to purchase any additional functions for the SAN Volume Controller. This function can also be used to migrate from a fully allocated volume to a thin-provisioned volume.

When using volume mirroring, the zero detect feature for thin-provisioned volumes allows you to reclaim unused allocated disk space (zeros) when converting a fully allocated volume to a thin-provisioned volume.

Using SAN Volume Controller mirrored volumes feature for data migration involves the following basic steps:

1. Add the target copy for source volume.
2. Run synchronization and wait for it to complete.
3. Remove the source volume.

**Important:** When using SAN Volume Controller mirroring volume features, the data migration can be stopped at any time without compromising data consistency on the primary volumes.

7.7.1 Environment description

Figure 7-17 on page 238 shows the environment used during testing. Operating system environments are not covered because if SAN Volume Controller is already in place, the procedure is the same for any operating system.

The example uses the following components and tools:

1. Windows Server 2008 as application host
2. IBM SAN Volume Controller with code release 6.2.0.0
3. Two backend disk subsystems (source and target)
4. Iometer to simulate read/write operations to logical volume during mirroring
5. Windows Performance monitor to track disk performance and availability from the host side
The SAN Volume Controller side has a storage pool created from an old disk subsystem called ITSO_old. The target storage pool is on a new disk subsystem and called ITSO_new.

From a migration point of view, the process is similar to the one described in 7.6, “SAN Volume Controller Volume migration between two storage pools” on page 227. However, the two volumes are kept in synchronization after migration is complete.

The SVCs mirrored volume background architecture has the following characteristics:

- By default, reads will still be serviced from the original volume, but this setting can be changed after the two volumes are synchronized.
- All writes are sent to both volumes synchronously.
- It is possible to split the two volumes at a defined point in time. This characteristic allows you to schedule a controlled split, such as when the host is powered off.
- Allows a defined point in time copy to be taken.
- The second copy of the volume can be removed at any time, allowing easy regression.
Figure 7-18 shows the storage pools on SAN Volume Controller ITSO1 as shown in the GUI.

![Image of SAN Volume Controller ITSO1 storage pools]

**Figure 7-18** SAN Volume Controller ITSO1 storage pools

Example 7-33 shows the storage pools on SAN Volume Controller ITSO1 shown in the command-line interface.

**Example 7-33** SAN Volume Controller ITSO1 storage pools from CLI

```
IBM_2145:ITSO1:superuser>svcinfo lsmdiskgrp
id name      status mdisk_count vdisk_count capacity extent_size free_capacity
            virtual_capacity used_capacity real_capacity
0  ITSO__old online 3           1           45.00GB  256         35.00GB 10.00GB          10.00GB       10.00GB
1  ITSO__new online 4           1           62.50GB  256         52.50GB 10.00GB          10.00GB       10.00GB
```

Figure 7-19 shows the SAN Volume Controller ITSO1 volumes from the GUI.

![Image of SAN Volume Controller volume information]

**Figure 7-19** SAN Volume Controller volume information

Example 7-34 shows the storage pools on SAN Volume Controller ITSO1 shown in the command-line interface.

**Example 7-34** SAN Volume Controller ITSO1 volumes from CLI

```
IBM_2145:ITSO1:superuser>svcinfo lsvdisk
id name          IO_group_id IO_group_name status capacity vdisk_UID
0  sle_5v4_vol01 0           io_grp0       online 10.00GB 60050768018106209800000000000003
```

Chapter 7. SAN Volume Controller-based migration
Example 7-35 shows host system with the volume mapped from old storage using SAN Volume Controller.

**Example 7-35  SAN Volume Controller volume mapped to host.**

```
C:\Program Files\IBM\SDDDSM>datapath.exe query device
Total Devices : 1
DEV#:    0  DEVICE NAME: Disk1 Part0  TYPE: 2145       POLICY: OPTIMIZED
SERIAL: 60050768018106209800000000000003
```

<table>
<thead>
<tr>
<th>Path#</th>
<th>Adapter/Hard Disk</th>
<th>State</th>
<th>Mode</th>
<th>Select</th>
<th>Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Scsi Port1 Bus0/Disk1 Part0</td>
<td>OPEN</td>
<td>NORMAL</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>Scsi Port1 Bus0/Disk1 Part0</td>
<td>OPEN</td>
<td>NORMAL</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>Scsi Port1 Bus0/Disk1 Part0</td>
<td>OPEN</td>
<td>NORMAL</td>
<td>114317240</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>Scsi Port1 Bus0/Disk1 Part0</td>
<td>OPEN</td>
<td>NORMAL</td>
<td>114315945</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>Scsi Port2 Bus0/Disk1 Part0</td>
<td>OPEN</td>
<td>NORMAL</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>Scsi Port2 Bus0/Disk1 Part0</td>
<td>OPEN</td>
<td>NORMAL</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>Scsi Port2 Bus0/Disk1 Part0</td>
<td>OPEN</td>
<td>NORMAL</td>
<td>114287066</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>Scsi Port2 Bus0/Disk1 Part0</td>
<td>OPEN</td>
<td>NORMAL</td>
<td>114288570</td>
<td>0</td>
</tr>
</tbody>
</table>

To simulate I/O load on the host side, Iometer created constant I/O load on the volume used for migration. Figure 7-20 shows the Iometer window with the simulated I/O load.
Figure 7-21 shows that the I/O load exists on the SAN Volume Controller side.

Figure 7-21   SAN Volume Controller performance view

**Tip:** The steps described in this section are the same regardless of the host operating system.

### 7.7.2 Creating mirrored volumes using the SAN Volume Controller GUI

To create mirrored volumes using the GUI, perform the following steps:

1. Click **Volumes** → **All Volumes**.
2. Right-click the volume and select **Volume Copy Actions** → **Add Mirrored Copy** (Figure 7-22).
3. Select **ITSO_new** as shown on Figure 7-23 and click **Add Copy**.

![Add Volume Copy - s1e_5v4_vol01 (ITSO_old)](image)

**Figure 7-23** SAN Volume Controller Mirrored Copy confirmation window

**Tip:** You can select Thin Provisioning option on this window.

**Figure 7-24** shows volume copies after adding mirrored copy.

![ITSO > Volumes > All Volumes](image)

**Figure 7-24** SAN Volume Controller volume with mirrored copy

**Note:** The asterisk (*) shows that Copy1 is the primary volume.
After you add the mirrored copy, the synchronization process starts immediately (Figure 7-25). When the synchronization finishes, you can change the primary copy volume from Copy1 to Copy0.

4. When synchronization is complete, click **Volumes → All Volumes**.
5. Right-click **Copy0** and select **Make Primary** as shown on Figure 7-26.

Volume **sle_5v4_vol01** changes storage pool location from ITSO_old to ITSO_new, but retains the same UID number as shown in Figure 7-27.

6. Click **Volumes → All Volumes**.
7. Right-click **Copy1** from the ITSO_old storage pool and select **Delete this Copy** as shown on Figure 7-28.

![Figure 7-28: SAN Volume Controller deleting mirrored volumes](image)

8. Click **OK** to confirm the deletion.

The migration process is complete and you are using the new disk subsystem. Figure 7-29 shows the status of volume `sle_5v4_vol01`, which is placed on the new storage pool called `ITSO_new`.

![Figure 7-29: SAN Volume Controller volume after migration](image)

**Tip:** SAN Volume Controller supports migration from virtualized MDisks and from image mode MDisks.

### 7.7.3 Creating mirrored volumes using CLI

This section addresses all procedures to migrate data using CLI commands. To do so, perform the following steps:

1. Add a mirrored volume copy for existing volume `sle_5v4_vol01` using the `addvdiskcopy` command (Example 7-36). Adding a copy to an existing volume changes a non-mirrored volume into a mirrored volume.

   **Example 7-36: SAN Volume Controller adding mirrored volume copy**

   ```
   IBM_2145:ITSO1:superuser>svcinfo lsvdisk
   id  name  IO_group_id IO_group_name  status  mdisk_grp_id mdisk_grp_name  capacity  type  FC_id  FC_name  RC_id  RC_name  vdisk_UID  fc_map_count  copy_count  fast_write_state  se_copy_count
   0  sle_5v4_vol01 0  io_grp0  online  0  ITSO_old 10.00GB  striped  60050768018106209800000000000001 0 1  not_empty 0
   
   IBM_2145:ITSO1:superuser>lsmdiskgrp
   id  name  status  mdisk_count  vdisk_count  capacity  extent_size  free_capacity  virtual_capacity  used_capacity  real_capacity  overallocation  warning  easy_tier  easy_tier_status
   0  ITSO_old  online  3  1  45.00GB  256  35.00GB  10.00GB  10.00GB  10.00GB  22  80  auto  inactive
   ```
1 ITSO_new online 4 0 62.50GB 256 62.50GB 0.00MB 0.00MB 0.00MB 0 80 auto inactive

IBM_2145:ITSO1:superuser>svctask addvdiskcopy -mdiskgrp 1 0
Vdisk [0] copy [0] successfully created

IBM_2145:ITSO1:superuser>svcinfo lsvdiskcopy
vdisk_id vdisk_name copy_id status sync primary mdisk_grp_id mdisk_grp_name
capacity type se_copy easy_tier easy_tier_status
0 sle_5v4_vol01 0 online no no 1 ITSO_new 10.00GB striped no on inactive
0 sle_5v4_vol01 1 online yes 0 0 ITSO_old 10.00GB striped no on inactive

IBM_2145:ITSO1:superuser>lsvdisksyncprogress
vdisk_id vdisk_name copy_id progress estimated_completion_time
0 sle_5v4_vol01 0 11 110719235627

IBM_2145:ITSO1:superuser>svctask chvdisk -primary 0 0

2. When the synchronization process is finished, change the primary volume from the ITSO_old storage pool to ITSO_new as shown in Example 7-37.

Example 7-37 SAN Volume Controller change primary copy

IBM_2145:ITSO1:superuser>lsvdiskcopy
vdisk_id vdisk_name copy_id status sync primary mdisk_grp_id mdisk_grp_name
capacity type se_copy easy_tier easy_tier_status
0 sle_5v4_vol01 0 online yes yes 1 ITSO_new 10.00GB striped no on inactive
0 sle_5v4_vol01 1 online yes no 0 ITSO_old 10.00GB striped no on inactive

IBM_2145:ITSO1:superuser>svctask chvdisk -primary 0 0

3. Delete the copy from the storage pool ITSO_old as shown in Example 7-38.

Example 7-38 SAN Volume Controller change primary copy

IBM_2145:ITSO1:superuser>lsvdiskcopy
vdisk_id vdisk_name copy_id status sync primary mdisk_grp_id mdisk_grp_name
capacity type se_copy easy_tier easy_tier_status
0 sle_5v4_vol01 0 online yes yes 1 ITSO_new 10.00GB striped no on inactive
0 sle_5v4_vol01 1 online yes no 0 ITSO_old 10.00GB striped no on inactive

IBM_2145:ITSO1:superuser>svctask rmvdiskcopy -copy 1 0

7.7.4 Performance analysis

Figure 7-30 shows the impact during the migration process. The performance is normal during migration phase except two short performance peaks. One is when you add the mirrored copy, and the second while changing the primary volume.

This example is from a testing environment. In a production environment, pre-migration planning is crucial for successful migration.

![Performance View](image)

Figure 7-30 Performance view

7.8 Data migration using SAN Volume Controller Metro Mirror

The SAN Volume Controller Metro Mirror can be used to migrate data while the application server is still active on the source storage (through the source SAN Volume Controller). The source SAN Volume Controller cluster in a Metro Mirror partnership is called the **Master SAN Volume Controller cluster**. The target SAN Volume Controller cluster is called the **Auxiliary SAN Volume Controller cluster**. This section outlines the steps to set up a Metro Mirror partnership, relationships and, optionally, Metro Mirror consistency groups.

In this example, a Metro Mirror relationship between two SAN Volume Controller clusters is used to demonstrate how the copy services function can migrate data between two physically separate locations. You can also use Metro Mirror as a data migration solution within a single SAN Volume Controller cluster.
Chapter 7. SAN Volume Controller-based migration

Figure 7-31 shows the test environment.

![SAN Volume Controller Metro Mirror Lab Setup](image)

### 7.8.1 SAN Volume Controller Metro Mirror partnership

The first step is to create a Metro Mirror partnership between the clusters that you want to migrate between. This section contains the steps for the following procedures:

- Creating a Metro Mirror partnership using the SAN Volume Controller GUI
- Creating a Metro Mirror partnership using the SAN Volume Controller CLI

**Creating a Metro Mirror partnership using the SAN Volume Controller GUI**

Create an SAN Volume Controller Metro Mirror partnership between the two clusters using the following steps:

1. Click **Copy Services → Partnerships → New Partnership**.
2. On the Create Partnership window, define the remote cluster as shown in Figure 7-32 on page 248. In this example, there is a single remote cluster named ITSO2. Select the cluster, specify the wanted bandwidth in MBps, and then click **Create**.
3. The final window shows a listing of the partnership just created as shown in Figure 7-33. The status of the partnership is **Fully Configured**, which means that the partnership has been created from both clusters. If this cluster is the first cluster to define the partnership, the status is listed as **Partially Configured**. Repeat this process from the other cluster to get to the **Fully Configured** state.

**Important:** Set the bandwidth used by the background copy process to less than or equal to the bandwidth that can be sustained by the communication link between the systems. The link must be able to sustain any host requests and the rate of background copy.

If the `-bandwidth` parameter is set to a higher value than the link can sustain, the background copy process uses the actual available bandwidth.
Creating a Metro Mirror partnership using the SAN Volume Controller CLI

You can also use SAN Volume Controller CLI commands to create the Metro Mirror partnership using the following steps:

1. Verify that the source and target SAN Volume Controller clusters can communicate by issuing the `svcinfo lsclustercandidate` command as shown in Example 7-39. In this example, the command is issued from the cluster named ITSO1. The candidate cluster name is ITSO2.

   **Example 7-39  List the candidate clusters**

   ```
   IBM_2145:ITSO1:admin>svcinfo lsclustercandidate
   id       configured cluster_name
   00000200604187DE no ITSO2
   IBM_2145:ITSO2:superuser>svcinfo lsclustercandidate
   id       configured name
   0000020060418826 yes ITSO1
   ```

2. To create the partnership between the two clusters, run the `svctask mkpartnership` command from each cluster (Example 7-40). The bandwidth parameter is mandatory.

   **Example 7-40  Create partnership**

   ```
   IBM_2145:ITSO1:superuser>svctask mkpartnership -bandwidth 50 00000200604187DE
   IBM_2145:ITSO2:superuser>svctask mkpartnership -bandwidth 50 0000020060418826
   ```

3. Verify the creation of the partnership using the `svcinfo lscluster` command as shown in Example 7-41. The partnership is displayed as partially configured if only one SAN Volume Controller cluster has created the partnership.

   **Example 7-41  Verify the SAN Volume Controller partnership**

   ```
   IBM_2145:ITSO1:superuser>svcinfo lscluster
   id       location partnership      bandwidth id_alias
   0000020060418826 local 0000020060418826
   00000200604187DE remote fully_configured 50 00000200604187DE
   IBM_2145:ITSO2:superuser>svcinfo lscluster
   id       location partnership      bandwidth id_alias
   00000200604187DE local 00000200604187DE
   0000020060418826 remote fully_configured 50 0000020060418826
   ```

### 7.8.2 SAN Volume Controller Metro Mirror relationships

After you establish the Metro Mirror partnership, you are ready to create the mirror relationships. This section contains the steps for the following procedures:

- Creating Metro Mirror relationships using SAN Volume Controller GUI
- Creating Metro Mirror relationships using SAN Volume Controller CLI
Creating Metro Mirror relationships using SAN Volume Controller GUI

To create the Metro Mirror relationships using the GUI, perform the following steps:

1. Click **Copy Services → Remote Copy → New Relationship**.
2. On the New Relationship window, select **Metro Mirror** and click **Next** (Figure 7-34).

![Figure 7-34 Steps for creating Metro Mirror relationships](image)

3. Select the Auxiliary Cluster ITS002 (on another system) where the volumes are located and click **Next** (Figure 7-35).

![Figure 7-35 Select the auxiliary cluster](image)
4. Select the master and auxiliary volume pair for the Metro Mirror. The master and auxiliary volume must be the same size. Only volumes that are suitable for the relationship are listed. In this example, select ITSO_Auxiliary as shown in Figure 7-36. Click **Add** and then **Next**.

![New Relationship](image)

*Figure 7-36   Select the Auxiliary Volume*

5. Select a synchronization option for the volumes (Figure 7-37). In this example, select **No** because this is the status of the relationship. Click **Next**.

![New Relationship](image)

*Figure 7-37   Select synchronization status*

6. Select **Yes** to start replication immediately (Figure 7-38), and click **Next**.

![New Relationship](image)

*Figure 7-38   Select to start copying process*
7. Track the synchronization process from the GUI by clicking **Running task** from the lower right window as shown in Figure 7-39.

![Figure 7-39 Remote Copy progress](image)

Creating Metro Mirror relationships using SAN Volume Controller CLI

When creating a relationship using the SAN Volume Controller CLI, you can create two different types of Metro Mirror relationships:

- **Associated with a consistency group**: Consistency groups preserve data consistency across multiple Metro Mirrored volumes. Use them when your applications have related data that span multiple volumes.
- **Stand-alone relationships**: Relationships can be created outside a consistency group.

**Creating relationships as part of a consistency group**

To create a Metro Mirror relationship within a consistency group, perform the following steps:

1. Create the consistency group using the `svctask mkrcconsistgrp` command (Example 7-42). In this example, the group is named `Lnxsmallfs_cg` and is being issued on local cluster ITSO1. The remote cluster is ITSO2. The `svcinfo lsrcconsistgrp` command verifies creation.

**Example 7-42  Create an empty consistency group**

```
IBM_2145:ITSO1:superuser>svctask mkrcconsistgrp -name Lnxsmallfs_cg -cluster ITSO2
RC Consistency Group, id [0], successfully created
```

```
IBM_2145:ITSO1:superuser>svcinfo lsrcconsistgrp
id name          master_cluster_id master_cluster_name aux_cluster_id aux_cluster_name primary state relationship_count copy_type
0  Lnxsmallfs_cg 0000020060418826  ITSO1               00000200604187DE ITSO2 empty 0                  empty_group
```

2. After a consistency group is created, create Metro Mirror relationships as part of that consistency group. Example 7-43 shows the Metro Mirror relationship named `lnx_smallfs_mm`, created as part of consistency group `Lnxsmallfs_cg`. It is between volume `ITSO_Master` on the source cluster and `ITSO_Auxiliary` on the target cluster. The command `svctask mkrcrelationship` creates the Metro Mirror relationship and `svcinfo lsrcrelationship` verifies its creation.

**Example 7-43  Create a Metro Mirror relationship associated with a consistency group**

```
IBM_2145:ITSO1:superuser>svctask mkrcrelationship -master ITSO_Master -aux ITSO_Auxiliary -name lnx_smallfs_mm -consistgrp Lnxsmallfs_cg -cluster ITSO2
```
Creating stand-alone relationships

If the Metro Mirror relationships do not need to be part of a consistency group, stand-alone relationships can be created using the same `svctask rmkrcrelationship` command. The `-consistgrp` parameter can be omitted in this case.

Example 7-44 shows the creation of a stand-alone relationship named `lnx_mediumfs_mm` between volumes `ITSO_Master` and `ITSO_Auxiliary`.

Example 7-44  Create a stand-alone Metro Mirror relationship

```
IBM_2145:ITSO1:superuser>svctask mkrcrelationship -master ITSO_Master -aux ITSO_Auxiliary -name lnx_mediumfs_mm -cluster ITSO2
RC Relationship, id [1], successfully created

IBM_2145:ITSO1:superuser>svcinfo lsrcrelationship 1
id 1
name lnx_mediumfs_mm
master_cluster_id 0000020060418826
master_cluster_name ITSO1
master_vdisk_id 1
master_vdisk_name ITSO_Master
aux_cluster_id 00000200604187DE
aux_cluster_name ITSO2
aux_vdisk_id 0
aux_vdisk_name ITSO_Auxiliary
primary master
consistency_group_id 0
consistency_group_name Lnxsmallfs_cg
state inconsistent_stopped
bg_copy_priority 50
progress 0
freeze_time
status online
sync
copy_type metro
```
7.8.3 Starting and monitoring SAN Volume Controller Metro Mirror Copy

The next step in the data migration is to copy the data to the target disk. This section contains the steps for the following procedures:

- Starting Metro Mirror Copy using SAN Volume Controller GUI
- Start Metro Mirror Copy using SAN Volume Controller CLI

Starting Metro Mirror Copy using SAN Volume Controller GUI

To copy the data using the SAN Volume Controller GUI, perform the following steps:

1. Click Copy Services → Remote Copy.
2. Select the Metro Mirror relationship (Figure 7-40). The state Inconsistent Stopped means that this relationship has not been started. In this example, the data on both Auxiliary volumes are not yet synchronized. Click Actions → Start.

![Figure 7-40 Start Copy process](image)

After the data is copied, the state for the two Volumes is Consistent Synchronized as shown in Figure 7-41.

![Figure 7-41 Consistent Synchronized state](image)

Start Metro Mirror Copy using SAN Volume Controller CLI

Metro Mirror consistency groups and stand-alone relationships are started and monitored with two sets of commands.
**Copying data in consistency groups**

Use the `svctask startrcconsistgrp` command to start the Metro Mirror copy for all the relationships in the consistency group (Example 7-45). The command `svcinfo lsrccconsistgrp` shows a state of inconsistent_copying. This state means that the copy is in progress. The command `svcinfo lsrcrelationships` shows the progress of the copy.

**Example 7-45  Start and monitor a Metro Mirror consistency group**

```
IBM_2145:ITS01:admin>svctask startrcconsistgrp Lnxsmallfs_cg

IBM_2145:ITS01:superuser>svcinfo lsrccconsistgrp Lnxsmallfs_cg
id 0
name Lnxsmallfs_cg
master_cluster_id 0000020060418826
master_cluster_name ITS01
aux_cluster_id 00000200604187DE
aux_cluster_name ITS02
primary master
state inconsistent_copying
relationship_count 1
freeze_time
status
sync
copy_type metro
RC_rel_id 1
RC_rel_name lnx_smallfs_mm

IBM_2145:ITS01:superuser>svcinfo lsrcrelationship lnx_smallfs_mm
id 1
name lnx_smallfs_mm
master_cluster_id 0000020060418826
master_cluster_name ITS01
master_vdisk_id 1
master_vdisk_name ITS0_Master
aux_cluster_id 00000200604187DE
aux_cluster_name ITS02
aux_vdisk_id 0
aux_vdisk_name ITS0_Auxiliary
primary master
consistency_group_id 0
consistency_group_name Lnxsmallfs_cg
state inconsistent_copying
bg_copy_priority 50
progress 9
freeze_time
status online
sync
copy_type metro
```

After all the pairs are synced, the consistency group state is consistent_synchronized (Example 7-46).

**Example 7-46  Consistency group that has been synchronized**

```
IBM_2145:ITS01:admin>svcinfo lsrccconsistgrp Lnxsmallfs_cg
id 0
```

Chapter 7. SAN Volume Controller-based migration  255
name Lnxsmallfs_cg
master_cluster_id 0000020060418826
master_cluster_name ITSO1
aux_cluster_id 00000200604187DE
aux_cluster_name ITSO2
primary master
state consistent synchronized
relationship_count 1
freeze_time
status sync
copy_type metro
RC_rel_id 1
RC_rel_name lnx_smallfs_mm

IBM_2145:ITSO1:superuser>svcinfo lsrcrelationship lnx_smallfs_mm
id 1
name lnx_smallfs_mm
master_cluster_id 0000020060418826
master_cluster_name ITSO1
master_vdisk_id 1
master_vdisk_name ITSO_Master
aux_cluster_id 00000200604187DE
aux_cluster_name ITSO2
aux_vdisk_id 0
aux_vdisk_name ITSO_Auxiliary
primary master
consistency_group_id 0
consistency_group_name Lnxsmallfs_cg
state consistent synchronized
bg_copy_priority 50
progress
freeze_time
status online
sync
copy_type metro

Copying data in stand-alone relationships
Stand-alone relationships are started using the svctask startrcrelationship command. Again, the progress of the copy can be monitored using the svcinfo lsrcrelationship command.

Example 7-47  Metro Mirror stand-alone relationship

IBM_2145:ITSO1:admin>svctask startrcrelationship lnx_mediumfs_mm

IBM_2145:ITSO1:superuser>svcinfo lsrcrelationship lnx_mediumfs_mm
id 1
name lnx_mediumfs_mm
master_cluster_id 0000020060418826
master_cluster_name ITSO1
master_vdisk_id 1
master_vdisk_name ITSO_Master
aux_cluster_id 00000200604187DE
aux_cluster_name ITSO2
aux_vdisk_id 0
aux_vdisk_name ITSO_Auxiliary
primary master
consistency_group_id
consistency_group_name
state inconsistent_copying
bg_copy_priority 50
progress 3
freeze_time
status online
sync
copy_type metro

IBM_2145:ITSO1:admin>svcinfo lsrcrelationship lnx_mediumfs_mm
id 1

name lnx_mediumfs_mm
master_cluster_id 0000020060418826
master_cluster_name ITSO1
master_vdisk_id 1
master_vdisk_name ITSO_Master
aux_cluster_id 00000200604187DE
aux_cluster_name ITSO2
aux_vdisk_id 0
aux_vdisk_name ITSO_Auxiliary
primary master
consistency_group_id
consistency_group_name
state consistent_synchronized
bg_copy_priority 50
progress
freeze_time
status online
sync
copy_type metro

7.8.4 Stopping SAN Volume Controller Metro Mirror Copy

After the mirror relationships reach a synchronized state, move the application to the new storage subsystem. There are two options available:

- Keep SAN Volume Controller in the configuration and connect the application to virtualized storage.
- Remove SAN Volume Controller and attach the application to the new storage subsystem.

After you migrate from the old storage to the new storage using SVC Metro Mirror, you must switch your application to the new storage. This requires the application to be shut down because you have to switch application access from old to new storage.

Before starting these steps, shut down the application and preparing the system to move to the new storage. In general, these steps include:

- Quiesce all I/O.
- Shut down the application server.
- Stop the Metro Mirror copy.
- Remove the Metro Mirror relationships.
- Start the application server on the new storage, with or without SAN Volume Controller.

**Stopping Metro Mirror using SAN Volume Controller GUI**

To stop Metro Mirror copy using the SAN Volume Controller GUI, perform the following steps:

1. From the Master cluster, click **Copy Services → Remote Copy**.
2. Select the consistency group (Lnxsmallfs_cg in this example) and click **Action → Stop** to stop Metro Mirror copy process as shown in Figure 7-42.

![Stop Copy Process](image1)

3. Enable write access to the target disk. Be sure to select **Allow secondary read/write access** as shown in Figure 7-43. Selecting this option allows the application server to access the target volumes as the new source volumes. Click **Stop Consistency Group** to stop the copy.

![Enable write access](image2)
The state of the relationship is now displayed as Idling. The relationship still exists, but no data is being copied. Figure 7-44 shows the status of the relationships in the example, In Sync/Idling, after the copy is stopped.

4. Select the consistency group (Lnxsmallfs_cg in the example) and click **Delete** as shown in Figure 7-45.

5. Click **OK** to confirm the deletion.
6. Delete the partnership between Master volume and Auxiliary volume by clicking **Copy Services → Remote Copy → Delete Partnership** as shown in Figure 7-46.

7. Click **OK** to confirm the deletion.

**Stopping Metro Mirror using SAN Volume Controller CLI**

To stop Metro Mirror copy using the SAN Volume Controller CLI, perform the following steps:

1. Stop the consistency groups using the `svctask stoprcconsistgrp` command with the `-access` parameter as shown in Example 7-48.

   **Example 7-48   Stop a consistency group and allow access to the target VDisk**

   ```
   IBM_2145:ITSO1:admin>svctask stoprcconsistgrp -access Lnxsmallfs_cg
   
   IBM_2145:ITSO1:superuser>svcinfo lsrcconsistgrp Lnxsmallfs_cg
   id 0
   name Lnxsmallfs_cg
   master_cluster_id 0000020060418826
   master_cluster_name ITSO1
   aux_cluster_id 00000200604187DE
   aux_cluster_name ITSO2
   primary
   state idling
   relationship_count 1
   freeze_time
   status
   sync in sync
   copy_type metro
   RC_rel_id 1
   RC_rel_name lnx_smallfs_mm
   ```
2. Stop the stand-alone Metro Mirror relationships and allow host access to the target volumes using the `svctask stoprcrelationship` command with the `-access` parameter (Example 7-49).

Example 7-49  Stop Metro Mirror relationships

```
IBM_2145:ITSO1:superuser>svctask stoprcrelationship -access lnx_smallfs_mm
IBM_2145:ITSO1:admin>svcinfo lsrcrelationship lnx_smallfs_mm
```

```
id 1
name lnx_smallfs_mm
master_cluster_id 0000020060418826
master_cluster_name ITSO1
master_vdisk_id 1
master_vdisk_name ITSO_Master
aux_cluster_id 00000200604187DE
aux_cluster_name ITSO2
aux_vdisk_id 0
aux_vdisk_name ITSO_Auxiliary
primary
consistency_group_id
consistency_group_name
state idling
bg_copy_priority 50
progress
freeze_time
status
sync in_sync
copy_type metro
```

### 7.8.5 Performance overview

This section addresses the performance impact during Metro Mirror synchronization for the example environment. A heavy I/O load was added during testing to simulate a real case scenario.

Figure 7-47 shows overall host System Summary during the Metro Mirror process.

![Figure 7-47](Image)
Figure 7-48 shows host Disk Busy graph.

![Disk Busy Graph]

The testing scenarios showed that Metro Mirror replication using SAN Volume Controller had no impact on the host I/O performance. This allowed sustained I/O load during Metro Mirror replication.

However, when planning for Metro Mirror replication, be sure that you use correct link sizing. Insufficient link bandwidth can saturate source disks. Contact your IBM technical representative to help you properly size the replication link.

### 7.9 SAN Volume Controller as data migration engine

The primary use of the SAN Volume Controller is not as a storage migration tool. However, the advanced capabilities of the SAN Volume Controller allow you to use the SAN Volume Controller to migrate data. Therefore, you can add the SAN Volume Controller temporarily to your SAN environment to copy the data from one storage subsystem to another storage subsystem.

The SAN Volume Controller allows you to copy image mode volumes directly from one subsystem to another subsystem while host I/O is running. The only downtime required is when the SAN Volume Controller is added to and removed from your SAN environment. This scenario is described in 7.5.6, “Preparing to migrate from the SAN Volume Controller” on page 220.

To use SAN Volume Controller for migration purposes only, perform the following steps:

1. Add SAN Volume Controller to your SAN environment.
2. Configure the SAN Volume Controller to fit your needs.
3. Prepare your application for data migration (unmount file systems and detach LUNs).
4. Add SAN Volume Controller between your storage and the host.
5. Create image mode disks on SAN Volume Controller from the LUNs that you migrate.
6. Attach LUNs, mount the file systems, and start the application.
7. Start the migration.
8. After the migration process is complete, detach the selected LUNs.
9. Remove SAN Volume Controller from your SAN.
10. Attach the LUNs, and start the application.

As you can see, little downtime is required. If you prepare everything correctly, you should be able to reduce your downtime to a few minutes. The copy process is handled by the SAN Volume Controller, so there is no performance impact on the host during the migration process.

7.10 Other resources

For more information about SAN Volume Controller based migration, see the following publications:

- *Implementing the IBM System Storage SAN Volume Controller V6.1*, SG24-7933
- *Software Installation and Configuration Guide Version 6.2.0*, GC27-2286-01
- Official SAN Volume Controller guides can be found at:
Chapter 8. Using mirroring techniques

This chapter addresses migrating data from a disk storage source device to a DS8000 target device using Logical Volume Manager (LVM)-based mirroring (split mirroring). This chapter contains the following sections:

- Concepts of the LVM
- Preparation and planning
- Migrating data using Windows Logical Disk Manager
- Data migration using Solaris Volume Manager
- Data migration using Veritas Storage Foundation
- Data migration using HP-UX Volume Manager mirroring
- Data migration using AIX LVM mirroring
- Data migration in a PowerHA clustered environment using AIX LVM mirroring
- Data migration using Linux LVM2 mirroring
- Data migration using network block devices
8.1 Concepts of the LVM

Logical Volume Manager (LVM) is available for most open systems. For AIX and Linux, it is called LVM. For Solaris and HP-UX, it can be known as LVM or VxVM. For the Windows platform, it is known as Logical Disk Manager (LDM). The LVM or LDM creates a layer of virtualization within the operating system.

Every volume management software provides the following basic functions:

- Extend logical volumes across several physical disks
- Stripe data across several physical disks to improve performance
- Mirror data for high availability and migration

The LUNs provided by a DS8000 are presented to the LVM or LDM as physical SCSI disks.

The normal process for a migration is to set up a mirror of the data on the old disks to the new LUNs. You wait until they are synchronized, and then split them at the cut-over time. Some volume management software provides commands that automate this process.

Disk mirroring was first mentioned in a 1978 patent awarded to Norman Ken Ouchi of the IBM Corporation. Logical Volume mirroring came into general use in UNIX operating systems around the mid 1990s. At that time, it was used primarily for data protection. However, the “split mirror” technique of moving data from one disk to another became a handy tool for the system administrator.

A lot has changed in the last 10 years, but LVM Mirroring is still an effective way to get data from one LUN to another. The process is straightforward. You have a logical volume on a LUN (data source) with data that you want to relocate or migrate to a LUN on a DS8000 (data destination). The process involves these steps:

1. Configure the destination LUN
2. Have the operating system recognize the LUN
3. Use the operating system LVM to create a logical volume on the DS8000 LUN
4. Establish a mirror between the source and destination logical volumes
5. Sync the mirror
6. Split the mirror

The data is now on the destination LUN.

The application using the source logical volume or file system does not need to be stopped during the LV mirror migration process. The process is non-disruptive as long as the operating system allows you to add and remove devices dynamically. However, typically you want the application to use the data on the destination logical volume after the migration. In this case, the application must be quiesced and reconfigured to use the destination logical volume. For a completely non-disruptive way to migrate data, use solutions such as those provided by Softek TDMF or zDMF.

A large advantage of using the volume management software for data migration is that it also allows for various types of consolidation. It can do this due to the virtualization nature of volume management software,
The major disadvantage of the volume management software mirroring method is that it requires substantial system administration, intervention, and attention. Production data is manipulated while production is running, and so migration requires host cycles.

**Important:** If you are planning to use volume management software functions to migrate data, be careful with limitations. These limitations include the total number of physical disks in the same volume group or volume set. In addition, if you are consolidating volumes with different sizes, check the procedures to see whether consolidation is possible.

### 8.2 Preparation and planning

As highlighted in Chapter 2, “Migration techniques and processes” on page 7, any migration project must start with a preparation and planning phase. Create a diagram of the overall environment (Figure 8-1 on page 269) and create a list of all the components in your environment.

Check the compatibility of the following components in your environment:

- Prepare the host servers. Verify that both the host and host HBAs can connect to the existing storage (source) and the DS8000 (target). See the Interoperability matrix available at:
  
  http://www.ibm.com/systems/support/storage/config/ssic/index.jsp

  - Check vendor and model type (change or add HBAs, if necessary).

  - Check the HBA firmware level and upgrade if necessary. Supported HBA firmware levels are available at the System Storage Interoperation Center (SSIC):
    
    http://www-03.ibm.com/systems/support/storage/ssic/interoperability.wss

  - Determine the number and size of each source volume (LUN) on each host.

  - Determine how the LUN is spread across multiple disks for performance considerations.

  - Determine whether the existing multipath driver can coexist with the supported host disk driver such as SDD, or MPIO.

- Prepare the DS8000 arrays and LUNs to provide the best possible performance.

  - Select RAID types, for example RAID-5 or RAID-10

  - Consider LUN to array mapping considerations for sharing, spreading, or isolating the type of requirements

  - Make sure that the latest firmware is loaded. You can verify the current level against the latest level at:
    

    http://www-03.ibm.com/systems/storage/software/virtualization/svc/interop.html

- Prepare the SAN fabric.

  - Check fabric compatibility for connection to both the existing storage server and the DS8000.
• Verify that fabric firmware levels are supported with the DS8000 (upgrade if necessary). For specific support, see the following Fibre Channel Director websites:
  Cisco SAN:
  http://www-1.ibm.com/servers/storage/san/c_type/
  CNT (Inrange) SAN:
  http://www-1.ibm.com/servers/storage/san/n_type/
  IBM SAN (Brocade) SAN:
  http://www-1.ibm.com/servers/storage/san/b_type/
  McData SAN:
  http://www-1.ibm.com/servers/storage/san/m_type/
• Verify that you have enough free ports to connect the DS8000 nodes to the fabric. Add more if needed.
  – Verify the cabling.
  • Order the correct number of cables.
  • Connect the DS8000 to the fabric.
  • Provide cabling diagrams to the cabling vendor or the person responsible for the cabling.
  – Verify the zoning.
  • Zone the host HBAs to the DS8000.
• Prepare the existing storage server.
  – Verify that the existing storage server is in optimal condition.
  • Check that all connections, ports, arrays, and LUNs are functioning properly.
  • Check firmware levels (upgrade if necessary).
  – Discover and document the logical and physical layout of the arrays and LUN mappings to the hosts.

8.3 Migrating data using Windows Logical Disk Manager

Dynamic disks were first introduced with the Windows 2000 server and provide features that basic disks do not. One of these features is the ability to create fault-tolerant volumes. This section addresses how to create a mirror using the Logical Disk Manager with dynamic disks.

8.3.1 Sample Windows LDM migration test environment

This section uses the test environment shown in Figure 8-1 on page 269. It is composed of the following components:
    – Disk device drivers:
      • Microsoft MPIO Version 6.0.6001.18000
      • IBM Subsystem Device Driver DSM Version 2.4.3.1-1
    – Emulex based LP1105 HBAs running firmware version 2.70A5
  ▶ IBM XIV running microcode level 10.2
DS8300 model 921 running firmware level SEA 5.4.3.65, bundle 6.2.400.76
- Brocade switch type 44, Fabric OS firmware version v6.3.0c

In this scenario, the XIV uses Microsoft Native MPIO and DS8000 LUNs managed by SDDDSM device drivers.

**Important**: Initially installing SDDDSM on the host server might require an outage if a reboot is needed.

The overall migration scenario is depicted in Figure 8-1. The three existing XIV data volumes (shown on the left side of the diagram in solid lines) are configured on the Windows server and are running live applications. The objective is to introduce the DS8000 into the environment and migrate the XIV volumes (source volumes) to the DS8000 volumes (target volumes). Three DS8000 volumes of equal or greater capacity using type ds are created in DS8000. The target volume capacities (sizes) are shown on the right side of the diagram.

![Figure 8-1 Migration test environment using Windows 2008 LDM](image)

### 8.3.2 High-level plan for migration using Windows LDM mirroring

The high-level steps needed to migrate using Windows LDM mirroring are shown in Table 8-1.

<table>
<thead>
<tr>
<th>Process</th>
<th>Commands</th>
<th>Explanation</th>
</tr>
</thead>
</table>
| Step 1. Identify the source (XIV) LUNs on the host server to be migrated. | 1. Right-click **My Computer**.  
2. Click **Manage**  
3. Click **Disk Management**. | This identification is done on the Windows host server. |
<table>
<thead>
<tr>
<th>Process</th>
<th>Commands</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 2.</strong> Assign the DS8000 volumes to the Windows host server and discover the new volumes.</td>
<td>1. <code>chvolgrp -dev image_id -action add -volume vol_num V#</code>&lt;br&gt;2. Click <em>My Computer.</em>&lt;br&gt;3. Click <em>Manage.</em>&lt;br&gt;4. Click <em>Disk Management.</em>&lt;br&gt;5. Click <em>Action</em> from the toolbar.&lt;br&gt;6. Click <em>Rescan disks.</em></td>
<td>Step 1 is done on the DS8000 with the dscii. You can choose to use the GUI. The <code>image_id</code> is the DS8000 id, and <code>vol_num</code> is the number of the DS8000 volume in hex. Steps 2-6 are performed on the Windows host server.</td>
</tr>
<tr>
<td><strong>Step 3.</strong> Bring the DS8000 target disks online.</td>
<td>1. Right-click a new disk&lt;br&gt;2. Click <em>Online.</em></td>
<td>This step is done on the Windows host server.</td>
</tr>
<tr>
<td><strong>Step 4.</strong> Identify the DS8000 LUN IDs.</td>
<td>1. Click Start&lt;br&gt;2. Select Subsystem Device DSM&lt;br&gt;3. Click Subsystem Device DSM&lt;br&gt;4. Enter the command <code>datapath query device</code> from the command prompt</td>
<td>This step is necessary if you have specific performance considerations. It is also needed if you have many LUNs that are the same size and you want to get LUN mapping details back to the DS8000 arrays.</td>
</tr>
<tr>
<td><strong>Step 5.</strong> Initiate the mirroring process.</td>
<td>1. Right-click the source volume (x:).&lt;br&gt;2. Click <em>Add Mirror.</em>&lt;br&gt;3. Select the chosen disk.&lt;br&gt;4. Click <em>Add Mirror.</em></td>
<td>The synchronization of the source and target volumes is automatic.</td>
</tr>
<tr>
<td><strong>Step 6.</strong> Verify that the volumes are copied and synced.</td>
<td>Visual check</td>
<td>Look for a disk that is functioning correctly.</td>
</tr>
<tr>
<td><strong>Step 7.</strong> Remove the mirror.</td>
<td>1. Right-click the source disk.&lt;br&gt;2. Click <em>Remove Mirror.</em>&lt;br&gt;3. Select the source disk to remove.&lt;br&gt;4. Click <em>Remove Mirror.</em>&lt;br&gt;5. Click <em>Yes.</em>&lt;br&gt;6. Right-click the selected volume.&lt;br&gt;7. Click <em>Remove Mirror.</em>&lt;br&gt;8. Click <em>Yes.</em></td>
<td>Make sure that you are removing the source disk from the mirror, not the DS8000 target disk.</td>
</tr>
<tr>
<td><strong>Step 8.</strong> Verify that the mirror is removed.</td>
<td>Visual check</td>
<td>Check that the disk is now deallocated.</td>
</tr>
<tr>
<td><strong>Step 9.</strong> Change the name of the new target disk to IDBM 2107.</td>
<td>1. Right-click the volume.&lt;br&gt;2. Click <em>Properties.</em>&lt;br&gt;3. Enter the volume name under the <em>General</em> tab.</td>
<td>Change the name of the new target disk to something meaningful.</td>
</tr>
<tr>
<td><strong>Step 10.</strong> Repeat steps 5-9.</td>
<td>N/A</td>
<td>Repeat the mirroring and renaming process for the remaining source and target disks.</td>
</tr>
<tr>
<td><strong>Step 11.</strong> Expand the DS8000 disks to their full capacity.</td>
<td>1. Right-click the volume.&lt;br&gt;2. Click <em>Extend Volume.</em>&lt;br&gt;3. Click the target disk.&lt;br&gt;4. Click <em>Next.</em>&lt;br&gt;5. Click <em>Finish.</em></td>
<td>In the example, the target disks are larger than the source LUNs, so you must expand them to use the remaining capacity.</td>
</tr>
</tbody>
</table>
8.3.3 Detailed steps for migration using Windows LDM mirroring

The procedure for doing the following steps is supplied for both the GUI and command line when possible.

**Identifying the source volumes on the Windows host server**

To identify the source volumes on the host server, perform the following steps:

1. From the Windows 2008 desktop on the host server, click the **Server Manager** icon.
2. In the Server Manager window, click **Storage** and then **Disk Management** as shown in Figure 8-2 on page 272.

**Tip:** You can also start Disk Management from the command line by typing **diskmgmt.msc**.

<table>
<thead>
<tr>
<th>Process</th>
<th>Commands</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 12.</strong> Delete the source volumes from the Windows host server.</td>
<td>1. Unassign the volume in the XIV box from the Windows host server. 2. Delete the zone in the fabric. 3. Disconnect the XIV storage device from the fabric or host server.</td>
<td>Follow the procedures for each component listed in the right column.</td>
</tr>
<tr>
<td><strong>Step 13.</strong> Rescan the Windows host server.</td>
<td>1. Right-click <strong>My Computer</strong>. 2. Click <strong>Manage</strong>. 3. Click <strong>Disk Management</strong>. 4. Click <strong>Action</strong> from the toolbar. 5. Click <strong>Rescan disks</strong>.</td>
<td>If a volume is displayed as missing, right-click the volumes and click <strong>Delete</strong>.</td>
</tr>
<tr>
<td><strong>Step 14.</strong> Verify that the device definitions are removed.</td>
<td>Visual check</td>
<td>Check that source volumes are gone.</td>
</tr>
</tbody>
</table>
3. Right-click the disk and select Properties.

4. Identify the disk manufacturer in the Properties window. In the example in Figure 8-2, the three volumes are Disk 2, 3, and 4. The drive letters x:, y:, and z: are associated with basic disks, which are the current source XIV volumes attached to the host.

**Tip:** To use the LDM function to mirror the LUNs, both the source and target disks need to be dynamic disks. For Windows 2008, you do not have to manually convert to dynamic disks because it is part of the mirroring process.
Assigning the DS8000 volumes to the Windows host server

Using the DS8000 dscli or DS GUI, assign the target LUNs to the host server using the following steps. Make sure that the target LUNs are the same size or larger than the source LUNs.

Important: Before proceeding, verify that you have properly cabled and zoned the connections from the DS8000 to the host server through the SAN fabric.

1. Assign the volumes, use the dscli command as shown in Example 8-1.

   **Example 8-1  Using the dscli commands to assign storage to the Windows host**

   ```
   mkfbvol -dev IBM.2107-7581981 -extpool P2 -name 8300_#h -type ds -cap 40 2000
   mkfbvol -dev IBM.2107-7581981 -extpool P2 -name 8300_#h -type ds -cap 50 2001
   mkfbvol -dev IBM.2107-7581981 -extpool P2 -name 8300_#h -type ds -cap 70 2002
   mkvolgrp -dev IBM.2107-75L4741 -type scsimap256 -volume 2000-2002 V6
   mkhostconnect -dev IBM.2107-7581981 -wwname 210000E08B875833 -hosttype Win2008
     -volgrp V6 x345-tic-17
   mkhostconnect -dev IBM.2107-7581981 -wwname 210000E08B09E5FD -hosttype Win2008
     -volgrp V6 x345-tic-17
   ```

2. Verify that the LUNs are indeed assigned to the volume group V6 by issuing the showvolgrp command as shown in Example 8-2.

   **Example 8-2  Using the showvolgrp command to verify the volumes in a volgrp from the dscli**

   ```
   dscli> showvolgrp -dev IBM.2107-75L4741 V6
   Date/Time: March 28, 2007 1:18:43 PM CST IBM DSCLI Version: 5.2.400.304 DS:
   IBM.2107-75L4741
   Name V6
   ID V6
   Type SCS1 Map 256
   Vols 2000 2001 2002
   ```
Windows 2008 automatically discovers the target LUNs. If the LUNs do not display on the Disk Management window, click **Action → Rescan** as shown in Figure 8-3.

![Figure 8-3  Running the Rescan tool on Windows](image-url)
The discovered LUNs are displayed as shown in Figure 8-4.

![Disk Management](image)

**Figure 8-4  View of disks after the Rescan**

In this example, Disks 4, 5, and 6 are the new DS8000 disks that will be part of the mirror (the target volumes). The disks are offline and not allocated.

3. Use command-line utility `diskpart` to rescan the disks as shown in Example 8-3.

**Example 8-3  Rescan disk using diskpart**

```
DISKPART> rescan
DISKPART> list disk
```

**Bringing the DS8000 target disks online**

Microsoft Windows 2008 introduced SAN policy to protect shared disks accessed by multiple servers. The default policy for Windows Advanced Server and Windows Data Center is OfflineShared. In this case, the boot disk and all disks that are not located on a shared bus.
such as SCSI, iSCSI, or SAS are brought online. The offline disks are read-only by default. On all other versions of Windows, the default is to bring all disks online. In this case, the disks are online and read/write.

You can use Diskpart utility to set the POLICY=OnlineAll to get around the default policy. However, if the disks are shared among servers, be aware that this setting can lead to data corruption. Use the correct SAN policy to protect your data.

After being brought online once, offline disks will be automatically online after the reboot.

To bring the target disks online, right-click one of the new disks and select Online as shown in Figure 8-5.

![Figure 8-5 Bringing disks online](image)

You can also use command-line utility Diskpart to make the disks online (Example 8-4).

**Example 8-4 Bringing a disk online with diskpart**

```
DISKPART> select disk 4
DISKPART> attributes disk clear readonly
DISKPART> online disk
```

**Identifying specific source and target LUNs**

If you have several disks of the same size and want to mirror specific volumes, you must identify the DS8000 volumes by the DS8000 LUN ID.
You can identify the LUNs by using the SDD command-line interface (SDDDSM) using the following steps:

1. From your Windows desktop, click **Start**.
2. Select **Subsystem Device DSM → Subsystem Device Driver DSM** (Figure 8-6).

![Figure 8-6 Using the SDDDSM CLI](image)

3. In the DSMCLI window, enter the command `datapath query device`.

The LUNs are shown in the red circles. The LUN IDs are represented by the last four digits of the serial numbers. For example, LUN 2000 at the top of the window is from LSS 20 and is the first LUN (00) from that LSS. The naming conventions were set so you know that LSS 20 is made from extent pool 2, and array 2 is located in extpool 2. Because each LUN is showing up through two paths, the SDD driver assigns one Dev number per two paths. For example, LUN 2000 is DEV 0, Disk 4.

**Note:** If you have specific requirements to share or isolate applications and files on separate arrays, map each LUN back to the DS8000. Spread or isolate these LUNs during the mirroring process.
Initiating the mirroring process

Now, with Disk 4, 5, and 6 brought online, the system is ready to initiate the mirroring process. To initiate the process, perform the following steps:

1. From the Disk Management window, right-click the source volume (x:) and select **Add Mirror** (Figure 8-7).

![Figure 8-7 Accessing Add Mirror window](image-url)
2. The Add Mirror window displays a list of available disks. Select the disk and click **Add Mirror** as shown in Figure 8-8.

![Figure 8-8 Selecting disks in the Add Mirror window](image)

3. In this example, the original disk is still a basic disk. A warning opens to confirm that you want to convert it to a dynamic disk as shown in Figure 8-9. Click **Yes** to continue.

![Figure 8-9 Warning of converting to dynamic disk](image)
After the mirror is added, the synchronization process starts automatically. At this time, you can see that both volumes, Disk 1 and Disk 4, are assigned to the same drive letter, x:

In Figure 8-10, the sync process is at 6%.

<table>
<thead>
<tr>
<th>Disk 1</th>
<th>Dynamic 32.00 GB Online</th>
<th>XIV (X:) 32.00 GB NTFS Resynching (6%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disk 2</td>
<td>Basic 48.00 GB Online</td>
<td>XIV (Y:) 48.00 GB NTFS Healthy (Primary Partition)</td>
</tr>
<tr>
<td>Disk 3</td>
<td>Basic 64.00 GB Online</td>
<td>XIV (Z:) 64.00 GB NTFS Healthy (Primary Partition)</td>
</tr>
<tr>
<td>Disk 4</td>
<td>Dynamic 40.00 GB Online</td>
<td>XIV (X:) 32.00 GB NTFS Resynching (6%)</td>
</tr>
<tr>
<td>Disk 5</td>
<td>Unknown 50.00 GB Not Initialized</td>
<td>50.00 GB Unallocated</td>
</tr>
<tr>
<td>Disk 6</td>
<td>Unknown 70.00 GB Not Initialized</td>
<td>70.00 GB Unallocated</td>
</tr>
</tbody>
</table>

Figure 8-10  Synchronization process running

You can also use command-line utility `Diskpart` to start the mirroring as shown in Example 8-5.

Example 8-5  Initiate the mirroring process with diskpart

```
DISKPART> select disk 1
DISKPART> convert dynamic
DISKPART> select disk 4
DISKPART> convert dynamic
DISKPART> select volume 2
DISKPART> add disk=4
```
Verifying that the volumes are copied and synchronized
Figure 8-11 shows the volumes after the synchronization process completes. Notice that the new volume is now healthy.

<table>
<thead>
<tr>
<th>Disk 1 Dynamic 32.00 GB Online</th>
<th>XIV (X:) 32.00 GB NTFS Healthy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disk 2 Basic 48.00 GB Online</td>
<td>XIV (Y:) 48.00 GB NTFS Healthy (Primary Partition)</td>
</tr>
<tr>
<td>Disk 3 Basic 64.00 GB Online</td>
<td>XIV (Z:) 64.00 GB NTFS Healthy (Primary Partition)</td>
</tr>
<tr>
<td>Disk 4 Dynamic 40.00 GB Online</td>
<td>XIV (X:) 32.00 GB NTFS Healthy</td>
</tr>
<tr>
<td>Disk 5 Unknown 50.00 GB Not Initialized</td>
<td>50.00 GB Unallocated</td>
</tr>
<tr>
<td>Disk 6 Unknown 70.00 GB Not Initialized</td>
<td>70.00 GB Unallocated</td>
</tr>
</tbody>
</table>

Figure 8-11  Synchronization process finished

You can also check the volume synchronization process with Diskpart as shown in Example 8-6. Notice that Volume 3 is rebuilding.

Example 8-6  Show the volume status with Diskpart

```
DISKPART> list volume

<table>
<thead>
<tr>
<th>Volume ###</th>
<th>Ltr</th>
<th>Label</th>
<th>Fs</th>
<th>Type</th>
<th>Size</th>
<th>Status</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volume 0</td>
<td>X</td>
<td>XIV</td>
<td>NTFS</td>
<td>Simple</td>
<td>40 GB</td>
<td>Healthy</td>
<td></td>
</tr>
<tr>
<td>Volume 1</td>
<td>C</td>
<td></td>
<td>NTFS</td>
<td>Partition</td>
<td>15 GB</td>
<td>Healthy</td>
<td>System</td>
</tr>
<tr>
<td>Volume 2</td>
<td>D</td>
<td>LAB-DATA</td>
<td>FAT32</td>
<td>Partition</td>
<td>2000 MB</td>
<td>Healthy</td>
<td></td>
</tr>
<tr>
<td>* Volume 3</td>
<td>Y</td>
<td>XIV</td>
<td>NTFS</td>
<td>Mirror</td>
<td>48 GB</td>
<td>Rebuild</td>
<td></td>
</tr>
<tr>
<td>Volume 4</td>
<td>Z</td>
<td>XIV</td>
<td>NTFS</td>
<td>Partition</td>
<td>64 GB</td>
<td>Healthy</td>
<td></td>
</tr>
</tbody>
</table>
```
Removing the mirror

In this step, you must choose one of two options:

- Break the mirrored volume: The selected volume keeps the original drive letter, and the other volume is automatically assigned another letter. The synchronization process no longer occurs. However, the data is retained.
- Remove the mirror: If you choose to remove the mirror, a window is displayed that asks you which volume you want to remove. The selected volume becomes a free disk with no drive letter assigned, and all data on it is erased.

In the example, the XIV source disk is no longer wanted in the environment, so you remove the mirror. To remove the mirror, perform these steps:

1. Right-click the disk that you want to remove the mirror and select **Remove Mirror** (Figure 8-12).

![Figure 8-12 Remove Mirror](image)

---

**Disk 1**
- Dynamic
- 32.00 GB
- Online
- XIV (X:)
- 32.00 GB NTFS
- Healthy
- Open
- Explore

**Disk 2**
- Basic
- 46.00 GB
- Online
- XIV (Y:)
- 46.00 GB NTFS
- Healthy (Primary Partition)
- Break Mirrored Volume...
- Change Drive Letter and Paths...
- Format...

**Disk 3**
- Basic
- 64.00 GB
- Online
- XIV (Z:)
- 64.00 GB NTFS
- Healthy (Primary Partition)
- Repair Volume...
- Reactivate Volume
- Delete Volume...

**Disk 4**
- Dynamic
- 41.00 GB
- Online
- XIV (X:)
- 32.00 GB NTFS
- Healthy
- Properties
- Help
2. In the Remove Mirror window, select the source XIV disk to remove as shown in Figure 8-13. In the example, Disk 1 is selected because it is the XIV source volume.

3. Click **Remove Mirror**.

4. Click **Yes** to confirm the removal.

You can also use **Diskpart** to remove the mirror as shown in Example 8-7.

**Example 8-7   Remove mirror with diskpart**

```bash
DISKPART> break disk=5 nokeep
DiskPart successfully broke the mirror volume.
```
Verifying that the mirror is removed

Verify that the selected volume is now available to the operating system without a drive letter or data as shown in Figure 8-14.

To verify that the mirror is removed with Diskpart, list the volume and check if the type is simple. If the mirror is not removed, the volume is listed as mirror as shown in Example 8-8.

Example 8-8  Verifying that the mirror removed with diskpart

```
DISKPART> list volume

<table>
<thead>
<tr>
<th>Volume</th>
<th>Ltr</th>
<th>Label</th>
<th>Fs</th>
<th>Type</th>
<th>Size</th>
<th>Status</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volume 0</td>
<td>X</td>
<td>IBM 2107</td>
<td>NTFS</td>
<td>Simple</td>
<td>40 GB</td>
<td>Healthy</td>
<td></td>
</tr>
<tr>
<td>Volume 1</td>
<td>C</td>
<td></td>
<td>NTFS</td>
<td>Partition</td>
<td>15 GB</td>
<td>Healthy</td>
<td>System</td>
</tr>
<tr>
<td>Volume 2</td>
<td>D</td>
<td>LAB-DATA</td>
<td>FAT32</td>
<td>Partition</td>
<td>2000 MB</td>
<td>Healthy</td>
<td></td>
</tr>
<tr>
<td>Volume 3</td>
<td>Y</td>
<td>XIV</td>
<td>NTFS</td>
<td>Simple</td>
<td>48 GB</td>
<td>Healthy</td>
<td></td>
</tr>
<tr>
<td>Volume 4</td>
<td>Z</td>
<td>XIV</td>
<td>NTFS</td>
<td>Partition</td>
<td>64 GB</td>
<td>Healthy</td>
<td></td>
</tr>
</tbody>
</table>
```
Changing the name of the new target disk

Change the name of the new target disk to something meaningful using the following steps:

1. Right-click the volume and select Properties.
2. Name the volume under the General tab as shown in Figure 8-15. In the example, the volume is called an IBM 2107 volume.

![Figure 8-15  Changing the volume properties](image)

Repeating the steps for the remaining disks

Repeat the steps starting with “Initiating the mirroring process” on page 278 for the remaining disks. You can mirror multiple disks in parallel.
Extending the target volumes

If you select target volumes that are larger than the source volumes, you need to extend the volumes to use the unused capacity on the target disks. To do extend the volumes, perform the following steps:

1. Right-click the volume and select **Extend Volume** as shown in Figure 8-16.

```
| Disk 1 | Basic 32.00 GB Online | 32.00 GB Unallocated |
| Disk 2 | Basic 40.00 GB Online | XIV (Y:) 45.00 GB NTFS Healthy (Primary Partition) |
| Disk 3 | Basic 64.00 GB Online | XIV (Z:) 64.00 GB NTFS Healthy (Primary Partition) |
| Disk 4 | Dynamic 40.00 GB Online | XIV (X:) 32.00 GB NTFS Healthy |
```

*Figure 8-16  View of extending the volumes to full capacity*
2. In the Extend Volume Wizard window, right-click the disk you want to extend and select **Extend Volume** (Figure 8-17).

![Extend Volume Wizard](image)

**Figure 8-17** View of the Extend Volume Wizard window

3. Click **Next**.

4. Click **Finish**.

The volume is now extended as shown in Figure 8-18.

![Extended Volume](image)

**Figure 8-18** View of extended volume

You can also extend the volume with **Diskpart** as shown in Example 8-9.

**Example 8-9** Extend volume with diskpart

```
DISKPART> select volume 1
Volume 1 is the selected volume.
DISKPART> extend
DiskPart successfully extended the volume.
```
Deleting the source volumes from the Windows host
After the data is migrated, you need to delete the original volumes. To do so, perform the following steps:

1. Unassign the volume in the source storage subsystem (XIV) from the Windows host server.
2. Delete the zone in the fabric.
3. Uncable (disconnect) the source (XIV) storage subsystem from the fabric or host server.

After this process is complete, visually verify that the device definitions are removed. If the volumes are displayed as missing, right-click the volumes and click Delete.

8.4 Data migration using Solaris Volume Manager

This section addresses the tasks required to migrate from two existing EMC LUNs to two new DS8000 LUNs using the Solaris Volume Manager (SVM) mirroring capabilities.

The process addressed here is fairly generic. The number of LUNs and the type of storage subsystem they originate from do not really matter. The process is the same for two or twenty LUNs from any subsystem. Depending on the specifics of your environment, the process can be disruptive to applications, requiring reboots and file system unmounts.

The example environment has the following characteristics:

- The data to be migrated is on two EMC LUNs.
- Each LUN has one logical volume that is on slice 2. Using slice 2 in Solaris allows you to cover the entire disk.
- Disk numbers 2 and 3 need to be migrated (Figure 8-19 on page 289) to DS8000 storage.
- The source volumes to be migrated are already under control of SVM and coded appropriately in /etc/vfstab.
- The source volumes are not SVM submirrors to an existing mirror.
- Because ensuring that the target LUNs are the same size as the source LUNs is difficult, the DS8000 LUNs are a little larger than the source LUNs.
- The same host system (Solaris) is attached to both the source and the destination storage subsystems.

8.4.1 Test environment

The test environment was composed of the following components:

- Sun Fire V280R host system
  - Solaris 10 11/06 (s10s_u3wos_10 SPARC) with Solaris Fibre Channel (FC) and Storage Multipathing software
  - Sun badged QLogic-based QLA2340 HBAs (SG-XPCI1FC-QL2) running firmware version 3.3.117
- EMC - 8730-18 - EMC Symmetrix 5 Storage running microcode level 5568.68
Figure 8-19 shows the hardware layout.

8.4.2 Solaris Volume Manager mirroring process

The Solaris Volume Manager mirroring process consists of the following basic steps:

1. Configure Solaris FC and multipathing software to recognize DS8000 storage: In this case, the server is equipped with Sun-badged HBAs. Sun HBAs allow you to use the Sun multipathing solution (Solaris Fibre Channel (FC) and Storage Multipathing software). SVM, other than Symantec VxVM, does not provide any built-in multipathing capability.

2. Determine source LUN information: This step needs to be done to discover which SVM object is associated with which device special file. You also need to know the size of the source LUNs to create appropriately sized target LUNs.

3. Discover the DS800 target LUNs.

4. Make the source volume a mirror member: This step might not be necessary if the source volumes are already a submirror to an existing mirror.

5. Make the source LUN SVM objects.

6. Make the destination volume a mirror member and synchronize the mirror: During this step, the source data is copied to the target devices.

7. Remove the source volumes from the mirror: This step removes the source LUNs from the SVM structures.
### 8.4.3 High-level plan for migration using SVM mirroring

The high-level steps are shown in Table 8-2.

<table>
<thead>
<tr>
<th>Process</th>
<th>Commands</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1.</strong> Configure Solaris Fibre Channel (FC) and Storage Multipathing software to recognize DS8000 storage.</td>
<td><code>vi /kernel/drv/scsi_vhci.conf</code></td>
<td>Open a vi session and update that file as shown in the detailed section.</td>
</tr>
<tr>
<td><strong>Step 2.</strong> Reboot the system.</td>
<td><code>stmsboot -u</code></td>
<td>Reboot to activate this change and enable multipathing for the DS8000 LUNs.</td>
</tr>
</tbody>
</table>
| **Step 3.** Determine information about the migration source LUNs. | 1. `more /etc/vfstab`  
2. `metastat d21` | Determine the association of the file system or mount points that are on the source volume with the underlying SVM object. You can determine the association by reviewing the contents of `/etc/vfstab` and using the `metastat` command. |
| **Step 4.** Discover the DS8000 migration destination LUNs. | `format` | Determine which LUNs belong to the DS8000. |
| **Step 5.** Make the source volume a mirror member. | 1. `metainit`  
2. `vi /etc/vfstab`  
3. `df -k` | This step is disruptive to the application because it requires the migration source volumes to be unmounted. If the source LUNs are already submirrors, this step can be omitted. Run the command `metainit`. Modify the `vsfstab`. Mount the mirrors. Verify the mount with `df -k`. |
| **Step 6.** Make the source LUNs SVM objects. | 1. `metainit d22 1 1`  
c5t6005076305FFC786000000000004001d0s2  
2. `metainit d32 1 1`  
c5t6005076305FFC78600000000000004003d0s1  
3. `metastat` | The next command sequence brings the DS8000 migration destination volumes under the control of Solaris SVM. They are designated as d22 and d32. |
| **Step 7.** Make the destination volume a mirror member and sync the mirror. | 1. `metattach d20 d22`  
2. `metattach d30 d32` | Create a two-way mirror by integrating the metadevice (d22) based on DS8000 into the already existing mirror (d20). |
| **Step 8.** Remove the source volumes from the mirror. | 1. `metadetach d20 d21`  
2. `metastat d20`  
3. `metadetach d30 d31`  
4. `metastat d30` | After the sync is complete, remove the mirror and verify with the `metastat` command. |
8.4.4 Detailed steps for migration using SVM mirroring

The migration procedure has the following detailed steps:

- Configuring Solaris FC and multipathing software for DS8000
- Rebooting the system
- Determining source LUN information
- Discovering the DS8000 target LUNs
- Making the source volume a mirror member
- Making the source LUNs SVM objects
- Making the destination volume a mirror member and synchronize
- Removing source volumes from the mirror and verifying the migration

Configuring Solaris FC and multipathing software for DS8000

To introduce a third-party symmetric disk subsystem to Solaris Fibre Channel (FC) and Storage Multipathing software, supply this subsystem information in the file `scsi_vhci.conf`:

- The vendor identification (VID)
- The product identification (PID) string

This file is in the `/kernel/drv` directory. The file needs to be updated as shown in Figure 8-20.

```bash
# vi /kernel/drv/scsi_vhci.conf
"/kernel/drv/scsi_vhci.conf" 31 lines, 1053 characters
#
# Copyright 2004 Sun Microsystems, Inc. All rights reserved.
# Use is subject to license terms.
#
#pragma ident "@(#)scsi_vhci.conf 1.9 04/08/26 SMI"
#
name="scsi_vhci" class="root";
#
load-balance="round-robin";
#
auto-failback="enable";
#
# For enabling MPxIO support for 3rd party symmetric device need an
# entry similar to following in this file. Just replace the "SUN SENA"
# part with the Vendor ID/Product ID for the device, exactly as reported by
# Inquiry cmd.
#
# device-type-scsi-options-list =
device-type-scsi-options-list =
# "SUN SENA", "symmetric-option";
"EMC SYMMETRIX", "symmetric-option",
"IBM 2107900", "symmetric-option";
#
# symmetric-option = 0x1000000;
symmetric-option = 0x1000000;
```

Figure 8-20  Enable DS8000 for Solaris multipath support
Rebooting the system

The system needs to be rebooted to activate the changes and enable multipathing for the DS8000 LUNs. Issue the command `stmsboot -u` as shown in Figure 8-21.

```bash
# stmsboot -u
WARNING: This operation will require a reboot.
Do you want to continue? [y/n] (default: y) y
The changes will come into effect after rebooting the system.
Reboot the system now? [y/n] (default: y) y
# stmsboot -u
WARNING: This operation will require a reboot.
Do you want to continue? [y/n] (default: y) y
```

Figure 8-21 Using stmsboot to enable multipathing for the DS8000 LUNs

Determining source LUN information

You need to gather information and understand several aspects about the migration source volumes and file systems. To do so, perform the following steps:

1. Determine the association of the file systems or mount points that are on the source volume with the underlying SVM object. Review the contents of the `/etc/vfstab` file and display the mounted file systems using the `df` command as shown in Example 8-10.

   In this example, the file system on metadevice `/dev/md/dsk/d21` is mounted on mount point `/emc08`. The file system on metadevice `/dev/md/dsk/d31` is mounted on mount point `/emc19`.

Example 8-10 Showing the association between mount points or file systems, and an SVM object

```bash
# more /etc/vfstab
#device         device          mount           FS      fsck    mount   mount
#to mount       to fsck         point           type    pass    at boot options
#fd      -       /dev/fd fd      -       no      -
/proc   -       /proc   proc    -       no      -
# non-mpxio: /dev/dsk/c2t0d0s1  -       -       swap    -       no      -
# mpxio: /dev/dsk/c5t20000004CFA3C467d0s1  -       -       swap    -       no      -
/dev/dsk/c5t20000004CFA3C467d0s1 -       -       swap    -       no      -
# non-mpxio: /dev/dsk/c2t0d0s0  /dev/rdsk/c2t0d0s0 / /       ufs     1       no      -
# mpxio: /dev/dsk/c5t20000004CFA3C467d0s0 /dev/rdsk/c5t20000004CFA3C467d0s0 /       ufs     1       no      -
/dev/dsk/c5t20000004CFA3C467d0s0 /dev/rdsk/c5t20000004CFA3C467d0s0 /       ufs     1       no      -
# non-mpxio: /dev/dsk/c2t0d0s7  /dev/rdsk/c2t0d0s7 /export/home ufs 2 yes     -
# mpxio: /dev/dsk/c5t20000004CFA3C467d0s7 /dev/rdsk/c5t20000004CFA3C467d0s7 /export/home ufs 2 yes     -
/export/home ufs 2 yes     -
/dev/dsk/c5t20000004CFA3C467d0s7 /dev/rdsk/c5t20000004CFA3C467d0s7 /export/home ufs 2 yes     -
/dev/md/dsk/d21 /dev/md/rdsk/d21 /emc08 ufs 2 yes     -
/dev/md/dsk/d31 /dev/md/rdsk/d31 /emc19 ufs 2 yes     -
/devices - /devices devfs - no      -
cvfs - /system/contract ctrfs - no      -
objfs - /system/object objfs - no      -
swap - /tmp tmpfs - yes     -
#
#
# df
/
   (/dev/dsk/c5t20000004CFA3C467d0s0): 3361778 blocks  482685 files
```
2. Discover the underlying LUNs of the metadevices by running the `metastat` command as shown in Example 8-11. In this example, the Solaris disk device `/dev/dsk/c5t6006048000028470097553594D304342d0s2` is associated with the SVM metadevice `d21`.

**Example 8-11  Issuing the metastat command**

```
# metastat d21
d21: Concat/Stripe
    Size: 17671680 blocks (8.4 GB)
    Stripe 0:
        Device                                             Start Block  Dbase  Reloc
/dev/dsk/c5t6006048000028470097553594D304342d0s2  0     No      Yes

Device Relocation Information:
Device Reloc Device ID
/dev/dsk/c5t6006048000028470097553594D304342d0  Yes  id1,ssd@n6006048000028470097553594d304342
```

3. Determine the size of the migration source LUNs so you know what size is required for the target LUN using the `format` command (Example 8-12).

**Example 8-12  Output of the format command**

```
# format
Searching for disks...done

c5t600604800002847009755643030432d0s2: configured with capacity of 6.56MB

AVAILABLE DISK SELECTIONS:
  0. c2t1d0 <SUN72G cyl 14087 alt 2 hd 24 sec 424>
```
In the sample environment, one unformatted EMC LUN with a device special file of c5t6006048000028470097556434D303030d0 and a size of 6.56 MB exists. It is the Volume Configuration Management Database LUN (VCMDB) that is used to store host attachment-specific information. It must not be used to store data.

4. Select the disk you want to migrate in the format menu (Example 8-13).

Example 8-13 Using the format dialog

```
# format
Searching for disks...done
```

```
c5t6006048000028470097556434D303030d0: configured with capacity of 6.56MB
c5t6005076305FFC786000000000004001d0: configured with capacity of 9.00GB
c5t6005076305FFC786000000000004002d0: configured with capacity of 9.00GB
c5t6005076305FFC786000000000004003d0: configured with capacity of 33.98GB
```

**AVAILABLE DISK SELECTIONS:**

```
0. c2t1d0 <SUN72G cyl 14087 alt 2 hd 24 sec 424>
1. c5t20000004CFA3C467d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
```

```
2. c5t6006048000028470097556434D303030d0 <EMC-SYMMETRIX-5568 cyl 14 alt 2 hd 15 sec 64>
3. c5t6006048000028470097553594D304342d0 <EMC-SYMMETRIX-5568 cyl 18408 alt 2 hd 15 sec 64>
```

```
4. c5t6006048000028470097553594D304533d0 <EMC-SYMMETRIX-5568 cyl 36818 alt 2 hd 30 sec 64>
5. c5t6006048000028470097553594D304638d0 <EMC-SYMMETRIX-5568 cyl 18408 alt 2 hd 15 sec 64>
```

Specify disk (enter its number):
8. Issue the partition command to print the partition table (Example 8-14).

Example 8-14 Using the partition dialog

selecting c5t6006048000028470097553594D304342d0 [disk formatted]

FORMAT MENU:
- disk: select a disk
- type: select (define) a disk type
- partition: select (define) a partition table
- current: describe the current disk
- format: format and analyze the disk
- repair: repair a defective sector
- label: write label to the disk
- analyze: surface analysis
- defect: defect list management
- backup: search for backup labels
- verify: read and display labels
- save: save new disk/partition definitions
- inquiry: show vendor, product and revision
- volname: set 8-character volume name
- !<cmd>: execute <cmd>, then return
- quit

format> pa

6. Issue the print command to print the partition information (Example 8-15). In the example environment, slice 2 of disk 3 is 8.43 GB. A slice in Solaris is a partition on the disk. Typically, slice 2 refers to the entire disk.

Example 8-15 Using the print dialog

format> pa

PARTITION MENU:
- 0: change '0' partition
- 1: change '1' partition
- 2: change '2' partition
- 3: change '3' partition
- 4: change '4' partition
- 5: change '5' partition
- 6: change '6' partition
- 7: change '7' partition
- select: select a predefined table
- modify: modify a predefined partition table
- name: name the current table
- print: display the current table
- label: write partition map and label to the disk
- !<cmd>: execute <cmd>, then return
- quit

Current partition table (original):
Total disk cylinders available: 18408 + 2 (reserved cylinders)

<table>
<thead>
<tr>
<th>Part</th>
<th>Tag</th>
<th>Flag</th>
<th>Cylinders</th>
<th>Size</th>
<th>Blocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>root</td>
<td>wm</td>
<td>0 - 273</td>
<td>128.44MB</td>
<td>(274/0/0)</td>
</tr>
<tr>
<td>1</td>
<td>swap</td>
<td>wu</td>
<td>274 - 547</td>
<td>128.44MB</td>
<td>(274/0/0)</td>
</tr>
<tr>
<td>2</td>
<td>backup</td>
<td>wu</td>
<td>0 - 18407</td>
<td>8.43GB</td>
<td>(18408/0/0)</td>
</tr>
<tr>
<td>3</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>(0/0/0)</td>
</tr>
<tr>
<td>4</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>(0/0/0)</td>
</tr>
<tr>
<td>5</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>(0/0/0)</td>
</tr>
<tr>
<td>6</td>
<td>usr</td>
<td>wm</td>
<td>548 - 18407</td>
<td>8.18GB</td>
<td>(17860/0/0)</td>
</tr>
<tr>
<td>7</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>(0/0/0)</td>
</tr>
</tbody>
</table>

Discovering the DS8000 target LUNs
In the example, two DS8000 migration destination LUNs were created on the DS8000. These LUNs are slightly larger than the source LUNs. Creating the target LUNs that are slightly larger is easier than trying to make them the same size. The DS8000 LUNs are one 9 GB LUN and one 33.9 GB LUN.

Note: DS8000 volumes for Solaris must be configured with type = scsimap256 and volume type = Sun.

Example 8-16 shows 5 disks/LUNs listed that are not labeled by Solaris. The DS8000 storage administrator had assigned twice the number of LUNs as requested. The LUN sizes are listed in the unlabeled section (first section).

Example 8-16  Output of the command format with DS8000 LUNs attached

# format
Searching for disks...done
c5t600604800002847097556434d030303d0: configured with capacity of 6.56MB
c5t6005076305f78600000000004010d0: configured with capacity of 9.00GB
c5t6005076305f78600000000004020d0: configured with capacity of 9.00GB
c5t6005076305f78600000000004000d0: configured with capacity of 9.00GB
c5t6005076305f78600000000004030d0: configured with capacity of 33.98GB

AVAILABLE DISK SELECTIONS:

Example 8-16  Output of the command format with DS8000 LUNs attached
The first LUN is the EMC storage that was noted previously. By comparing the device special file number in the unlabeled LUNs and the numbered LUNs shown in the AVAILABLE DISK SELECTIONS: section, you can determine which LUNs belong to the DS8000. See the disk list in the `format` command output in Example 8-13 on page 294.

Select the LUNs that you want to use as migration targets as shown at the bottom of Example 8-16 on page 296. In the example, LUNs 6 and 9 are chosen as the DS8000 migration targets.

Table 8-3 summarizes the source and target LUNs used in the scenario. It shows the native device special file names, the associated metadevice names, and the respective sizes.

<table>
<thead>
<tr>
<th>Source</th>
<th>Size of source (GB)</th>
<th>Target</th>
<th>Size of target (GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>c5t6006048000028470097553594D304342d0</td>
<td>8.43</td>
<td>c5t6005076305FFC78600000000004001d0</td>
<td>9.00</td>
</tr>
<tr>
<td>d21</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c5t6006048000028470097553594D304533d0</td>
<td>33.71</td>
<td>c5t6005076305FFC78600000000004003d0</td>
<td>33.98</td>
</tr>
<tr>
<td>d31</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Making the source volume a mirror member**

**Important:** This step is disruptive to the application because it requires the migration source volumes to be unmounted. If the source LUNs are already submirrors, this step can be omitted.

The command sequence shown in Example 8-17 includes the following steps:

1. Create a one-way mirror using the primary volume names (d20 and d30 in the example)
2. Make the source volumes (d21 and d31) submirrors.
3. Unmount the file systems on the disks that are to be migrated. You will remount them later, using the name of the mirror rather than the name of the submirror.

**Example 8-17** Creating a one-way mirror and taking DS8000 LUNs under control of SVM

```
# metainit d20 -m d21
d20: Mirror is setup
#
# metainit d30 -m d31
```
d30: Mirror is setup
#
# umount /emc08
# umount /emc19
#

Edit /etc/vfstab to reflect the new primary volume (d20 and d30) names. Example 8-18 shows what /etc/vfstab looks like after these changes are completed.

**Example 8-18  Mount the mirrors**

```bash
# vi /etc/vfstab
"/etc/vfstab" 20 lines, 1036 characters
#device         device          mount           FS      fsck    mount   mount
#to mount       to fsck         point           type    pass    at boot options
# fd - /dev/fd fd - no -
/proc - /proc proc - no -
# non-mpxio: /dev/dsk/c2t0d0s1 - - swap - no -
# mpxio: /dev/dsk/c5t20000004CFA3C467d0s1 - - swap - no -
/dev/dsk/c5t20000004CFA3C467d0s1 - - swap - no -
# non-mpxio: /dev/dsk/c2t0d0s0 /dev/rdsk/c2t0d0s0 / ufs 1 no -
# mpxio: /dev/dsk/c5t20000004CFA3C467d0s0 /dev/rdsk/c5t20000004CFA3C467d0s0 /
ufs 1 no -
/dev/dsk/c5t20000004CFA3C467d0s0 /dev/rdsk/c5t20000004CFA3C467d0s0 / ufs
1 no -
# non-mpxio: /dev/dsk/c2t0d0s7 /dev/rdsk/c2t0d0s7 /export/home ufs 2 yes -
# mpxio: /dev/dsk/c5t20000004CFA3C467d0s7 /dev/rdsk/c5t20000004CFA3C467d0s7
/export/home ufs 2 yes -
/dev/dsk/c5t20000004CFA3C467d0s7 /dev/rdsk/c5t20000004CFA3C467d0s7
/export/home ufs 2 yes -
/dev/md/dsk/d20 /dev/md/rdsk/d20 /emc08 ufs 2 yes -
/dev/md/dsk/d30 /dev/md/rdsk/d30 /emc19 ufs 2 yes -
/devices - /devices devfs - no -
ctfs - /system/contract ctfss - no -
objfs - /system/object objfs - no -
swap - /tmp tmpfs - yes -
#```

298  Data Migration to IBM Disk Storage Systems
Remount the file systems and verify that they are mounted with the `df` command as shown in Example 8-19.

**Example 8-19   Display file systems**

```
df -k
Filesystem            kbytes    used   avail capacity  Mounted on
...                    
/dev/md/dsk/d30      34809572  1298018  33163459   4%   /emc19
/dev/md/dsk/d20      8701901  7837153   777729   91%   /emc08
```

**Making the source LUNs SVM objects**

The next command sequence (Example 8-20) brings the DS8000 migration destination volumes under the control of Solaris SVM, They are designated as d22 and d32.

**Example 8-20   Making SVM objects out of the target LUNs**

```
# metainit d22 1 1 c5t6005076305FFC786000000000004001d0s2
d22: Concat/Stripe is setup
#
# metainit d32 1 1 c5t6005076305FFC786000000000004003d0s2
d32: Concat/Stripe is setup
#
```

Use the `metastat` command to show the hierarchal relationship between the primary volume and its submirror members as shown in Example 8-21. The metadevices created on the DS8000 LUNs (d22 and d32) do not belong to a mirror.

**Example 8-21   Using the command metastat to show the status of the mirror**

```
# metastat
```

```
d30: Mirror
   Submirror 0: d31
      State: Okay
      Pass: 1
      Read option: roundrobin (default)
      Write option: parallel (default)
      Size: 70690560 blocks (33 GB)

d31: Submirror of d30
   State: Okay
   Size: 70690560 blocks (33 GB)
   Stripe 0:
      Device                                             Start Block  Dbase        State
      /dev/dsk/c5t6006048000028470097553594D304533d0s2          0     No            Okay
      Yes

d20: Mirror
   Submirror 0: d21
      State: Okay
      Pass: 1
      Read option: roundrobin (default)
      Write option: parallel (default)
      Size: 17671680 blocks (8.4 GB)

d21: Submirror of d20
   State: Okay
   Size: 17671680 blocks (8.4 GB)
   Stripe 0:
```
<table>
<thead>
<tr>
<th>Device</th>
<th>Start Block</th>
<th>Dbase</th>
<th>State</th>
<th>Reloc</th>
<th>Hot Spare</th>
</tr>
</thead>
<tbody>
<tr>
<td>/dev/dsk/c5t6006048000028470097553594d030432d0s2</td>
<td>0</td>
<td>No</td>
<td>Okay</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>d32: Concat/Stripe</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Size: 71270400 blocks (33 GB)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stripe 0:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/dev/dsk/c5t6005076305FFC7860000000000004003d0s2</td>
<td>0</td>
<td>No</td>
<td>Reloc</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>/dev/dsk/c5t6005076305FFC7860000000000004001d0s2</td>
<td>0</td>
<td>No</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Device</th>
<th>Start Block</th>
<th>Dbase</th>
<th>State</th>
<th>Reloc</th>
<th>Hot Spare</th>
</tr>
</thead>
<tbody>
<tr>
<td>/dev/dsk/c5t6005076305FFC7860000000000004003d0s2</td>
<td>0</td>
<td>No</td>
<td>Okay</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>d22: Concat/Stripe</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Size: 18869760 blocks (9.0 GB)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stripe 0:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/dev/dsk/c5t6005076305FFC7860000000000004001d0s2</td>
<td>0</td>
<td>No</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Device Relocation Information:

<table>
<thead>
<tr>
<th>Device</th>
<th>Reloc</th>
<th>Device ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>/dev/dsk/c5t6005076305FFC78600000000004003d0s2</td>
<td>Yes</td>
<td>id1,ssd@n6005076305ffcc78600000000004003</td>
</tr>
<tr>
<td>/dev/dsk/c5t6005076305FFC78600000000004001d0s2</td>
<td>Yes</td>
<td>id1,ssd@n6005076305ffcc78600000000004001</td>
</tr>
<tr>
<td>/dev/dsk/c5t6006048000028470097553594d0304533d0s2</td>
<td>Yes</td>
<td>id1,ssd@n6006048000028470097553594d304533</td>
</tr>
<tr>
<td>/dev/dsk/c5t6006048000028470097553594d30432d0s2</td>
<td>Yes</td>
<td>id1,ssd@n6006048000028470097553594d30432</td>
</tr>
</tbody>
</table>

Making the destination volume a mirror member and synchronize

Create a two-way mirror by integrating the metadevice (d22) based on DS8000 into the already existing mirror (d20) using the `metattach` command. This command starts the synchronization process shown in Example 8-22. You can check the progress of the mirror synchronization using the `metastat` command.

**Example 8-22 Using metattach to create a two-way mirror and check the status**

```
# metattach d20 d22
d20: submirror d22 is attached
#
#
# metastat d20

d20: Mirror
    Submirror 0: d21
    State: Okay
    Submirror 1: d22
    State: Resyncing
    Resync in progress: 3 % done
    Pass: 1
    Read option: roundrobin (default)
    Write option: parallel (default)
    Size: 17671680 blocks (8.4 GB)

d21: Submirror of d20
    State: Okay
    Size: 17671680 blocks (8.4 GB)
    Stripe 0:
        Device                                             Start Block | Dbase | State |
        /dev/dsk/c5t6006048000028470097553594d030432d0s2 | 0     | No    | Okay  |
        Yes
```
d22: Submirror of d20
State: Resyncing
Size: 18869760 blocks (9.0 GB)
Stripe 0:
Device                                             Start Block  Dbase
State Reloc Hot Spare
/dev/dsk/c5t6005076305FFC786000000000004001d0s2          0     No
Okay   Yes
Device Relocation Information:
Device                                           Reloc  Device ID
/dev/dsk/c5t6006048000028470097553594D304342d0   Yes
id1,ssd@n6006048000028470097553594d304342
/dev/dsk/c5t6005076305FFC786000000000004001d0   Yes
id1,ssd@n6005076305ffc786000000000004001

Repeat the steps for the other mirror (d30) and submirror (d32) as shown in Example 8-23.

Example 8-23 Using metattach to create a two-way mirror and check the status
metattach d30 d32
 d30: submirror d32 is attached
 #
 # metastat d30
 d30: Mirror
 Submirror 0: d31
 State: Okay
 Submirror 1: d32
 State: Resyncing
 Resync in progress: 3 % done
 Pass: 1
 Read option: roundrobin (default)
 Write option: parallel (default)
 Size: 70690560 blocks (33 GB)

 d31: Submirror of d30
 State: Okay
 Size: 70690560 blocks (33 GB)
Stripe 0:
Device                                             Start Block  Dbase        State
Reloc Hot Spare
/dev/dsk/c5t6006048000028470097553594D304342d0          0     No            Okay
Yes

 d32: Submirror of d30
 State: Resyncing
 Size: 71270400 blocks (33 GB)
Stripe 0:
Device                                             Start Block  Dbase        State
Reloc Hot Spare
/dev/dsk/c5t6005076305FFC786000000000004003d0s2          0     No            Okay
Yes

Device Relocation Information:
Monitor the state of the destination volume submirror until the status is Okay. This status indicates that the mirror is in sync and complete as seen in Example 8-24.

### Example 8-24  Mirror synchronization is complete

```bash
# metastat d20
d20: Mirror
   Submirror 0: d21
      State: Okay
   Submirror 1: d22
      State: Okay
      Pass: 1
      Read option: roundrobin (default)
      Write option: parallel (default)
      Size: 17671680 blocks (8.4 GB)

d21: Submirror of d20
   State: Okay
   Size: 17671680 blocks (8.4 GB)
   Stripe 0:
      Device /dev/dsk/c5t6006048000028470097553594D30432d0s2
      Start Block 0
      Dbase No

Okay Yes

d22: Submirror of d20
   State: Okay
   Size: 18869760 blocks (9.0 GB)
   Stripe 0:
      Device /dev/dsk/c5t6005076305FFC786000000000004001d0s2
      Start Block 0
      Dbase No

Okay Yes
```
The current configuration is shown in Figure 8-22.

---

**Figure 8-22**   Entire setup after full mirroring is established

### Removing source volumes from the mirror and verifying the migration

The two-way mirror is now synchronized, so you can remove the original source volumes from the SVM. Use the `metadetach` command to detach the mirrors and submirrors as shown in Example 8-25. In the example, detach mirror d20 and submirror d21, and mirror d30 and submirror d31. You can use the `metastat` command to verify the results.

**Example 8-25**   Using `metadetach` to remove a metaset from a mirror

```
# metadetach d20 d21
# metastat d20
# metadetach d30 d31
```

---

This example demonstrates the use of `metadetach` and `metastat` commands to manage mirror and submirror detachments and verifications. It provides a practical example of how to manipulate mirror configurations in a specified SVM environment, ensuring data integrity and system stability during migration processes.
The migration is now complete, and all data is located exclusively on DS8000 LUNs. There might be further tasks remaining such as disconnecting the source LUNs from the system.

### 8.5 Data migration using Veritas Storage Foundation

This section addresses migrating data from three existing EMC LUNs to three DS8000 LUNs using the Veritas Volume Manager (VxVM) mirroring capabilities. This description is fairly generic. The number of LUNs does not really matter: The process is the same for three or more.

The example in this section describes an environment with the following characteristics:

- The data you want to migrate is on three EMC LUNs. It can, however, be on any other type of storage system without affecting the process described.
- Each source LUN has one logical volume on slice 2. Slice 2 in Solaris covers the entire disk.
- The EMC source volumes are already under control of VxVM and coded appropriately in /etc/vfstab. They are mounted on /emcN, 0 ≤ N ≤ 2.
- You are migrating the data from three source LUNs to three target LUNs on a DS8000. It is difficult to ensure that the target LUNs are the same size as the source LUNs. Therefore, use DS8000 LUNs are a little larger than the source LUNs.
- The same host system is attached to both the source and the destination storage subsystems.

#### 8.5.1 Test environment

The physical test environment is composed of the following components:

- Sun Fire V280R host system
  - Solaris 10 11/06 (s10s_u3wos_10 SPARC)
  - Sun-badged QLogic based QLA2340 HBAs (SG-XPCI1FC-QL2) running firmware version 3.3.117
8.5.2 Veritas Volume Manager mirroring

The Veritas Volume Manager mirroring technique has the following basic steps:

1. Install multipathing software: In the example, the server is equipped with Sun-badged HBAs. Sun HBAs allow use of the Sun multipathing solution (Solaris Fibre Channel (FC) and Storage Multipathing software). However, the example uses DMP for multipathing.

2. Determine migration source LUN information. This step needs to be done to discover which VxVM object is associated with which device special file. Beyond that you need to know the size of the source LUNs to provide target LUNs with an appropriate size.

3. Prepare the migration target volumes and investigate their VxVM status.

4. Put the target device under VxVM control.

5. Make the destination volume a mirror member and synchronize the mirror. This is that step where source data is copied to the target devices.

6. Remove source volumes from the mirror. This step needs to be done to get the source LUNs out of the VxVM structures.
8.5.3 High-level plan for migration using VxVM mirroring

The high-level steps are shown in Table 8-4.

Table 8-4 Example migration procedure using Solaris with Veritas Storage Foundation

<table>
<thead>
<tr>
<th>Process</th>
<th>Commands</th>
<th>Explanation</th>
</tr>
</thead>
</table>
| **Step 1.** Determine the source devices. | 1. format  
2. df  
3. vxprint  
4. vxdisk list EMC0_0 | Determine the size of the EMC migration source file systems controlled by VxVM and the migration source mounted file systems. Also look at the output of vxprint to prepare to associate objects with Solaris device files. |
| **Step 2.** Prepare the migration volumes. | 1. vxdisk list | Prepare the target LUNs to be used as mirror targets. Preparation includes verifying that the target LUNs are visible from the OS. Label them if not already done. |
| **Step 3.** Investigate the VxVM migration target device status. | 1. vxdisk list  
IBM_DS8x000_0  
2. format | Determining the DS8000 LUN size. |
| **Step 4.** Take IBM DS8000 LUNs under control of VxVM. | vxdiskadm | Take the target devices under control of VxVM, in this case using the CLI. |
| **Step 5.** Establish the mirrors. | 1. vxdiskadm  
2. vxtask list | Establish the mirrors using vxdiskadm from the CLI and monitor the progress. |
| **Step 6.** Remove the source storage from the mirrors. | 1. vxplex -g dg0 -o rm dis vol0-01  
2. vxplex -g dg1 -o rm dis vol1-01  
3. vxplex -g dg2 -o rm dis vol2-01  
4. vxprint | Remove the source LUNs from the mirrors. |
| **Step 7.** Remove the source LUNs from the VxVM disk groups. | 1. vxdg -g dg0 rmdisk dg001  
2. vxdg -g dg1 rmdisk dg101  
3. vxdg -g dg2 rmdisk dg201 | Remove the EMC LUNs from the disk group using vxdg -g <dgN> rmdisk. |
| **Step 8.** Verify that the LUN is removed from the disk group. | 1. vxdisk list  
2. vxprint | Display the migration status. |

8.5.4 Detailed steps for migration using VxVM mirroring

The migration procedure has the following detailed steps:

- Determining source LUN information
- Preparing the migration volumes
- Investigating the VxVM migration target device status
- Taking IBM DS8000 LUNs under control of VxVM
- Establishing the mirrors
- Removing the source LUNs from the mirrors
- Removing the source LUNs from the VxVM disk groups
Determining source LUN information

To determine the size of the DS8000 migration target volumes, determine the size of the migration source file systems controlled by VxVM using the following steps:

1. Enter the `format` command to display the LUNs that are currently visible as shown in Example 8-26.

   Twice as many LUNs show up than expected because DMP is a non-path suppressing software. This is different from Solaris Fibre Channel (FC) and Storage Multipathing software. Original devices are still seen even after installation of VxVM and DMP. Therefore, you see twice the number of disks (once through controller “c2” and once through controller “c4”) than are assigned to the operating system.

Determine which of these disks represent the migration source volumes and, most importantly, their size.

**Tip:** There are two paths representing the EMC LUNs (c2t500604843EOC4BCBd0 and c4t500604843EOC4BD4d0; size: 6.56 MB) that stay unformatted. These paths are the EMC Volume Configuration Management Database LUNs (VCMDB) that store host attachment-specific information. These paths must not be used to store data.

**Example 8-26  Display LUNs**

```
# format
Searching for disks...done

c2t500604843EOC4BCBd0: configured with capacity of 6.56MB

c4t500604843EOC4BD4d0: configured with capacity of 6.56MB

AVAILABLE DISK SELECTIONS:
  0. c1t0d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
     /pci@8,600000/SUNW,qlc@4/fp@0,0/ssl@w21000004cfa37b97,0

  1. c1t1d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
     /pci@8,600000/SUNW,qlc@4/fp@0,0/ssl@w21000004cfac3616,0

  2. c2t500604843EC4BCBd0 <EMC-SYMMETRIX-5568 cyl 14 alt 2 hd 15 sec 64>
     /pci@8,600000/fibre-channel@1/fp@0,0/ssl@w500604843e0c4bcb,0

  3. c2t500604843EC4BCBd9 <EMC-SYMMETRIX-5568 cyl 36818 alt 2 hd 30 sec 64>
     /pci@8,600000/fibre-channel@1/fp@0,0/ssl@w500604843e0c4bcb,9

  4. c2t500604843EC4BCBd20 <EMC-SYMMETRIX-5568 cyl 18408 alt 2 hd 15 sec 64>
     /pci@8,600000/fibre-channel@1/fp@0,0/ssl@w500604843e0c4bcb,14

  5. c2t500604843EC4BCBd29 <EMC-SYMMETRIX-5568 cyl 18408 alt 2 hd 15 sec 64>
     /pci@8,600000/fibre-channel@1/fp@0,0/ssl@w500604843e0c4bcb,1d

  6. c2t50050763050C0786d4 <IBM-2107900-.437 cyl 4316 alt 2 hd 64 sec 256>
     /pci@8,600000/fibre-channel@1/fp@0,0/ssl@w50050763050c0786,4

  7. c2t50050763050C0786d6 <IBM-2107900-.437 cyl 14 alt 2 hd 64 sec 256>
     /pci@8,600000/fibre-channel@1/fp@0,0/ssl@w50050763050c0786,6

  8. c2t50050763050C0786d8 <IBM-2107900-.437 cyl 14 alt 2 hd 64 sec 256>
     /pci@8,600000/fibre-channel@1/fp@0,0/ssl@w50050763050c0786,8

  9. c4t500604843EC4BD4d0 <EMC-SYMMETRIX-5568 cyl 14 alt 2 hd 15 sec 64>
     /pci@8,700000/fibre-channel@1/fp@0,0/ssl@w500604843e0c4bd4,0

  10. c4t500604843EC4BD4d7 <EMC-SYMMETRIX-5568 cyl 36818 alt 2 hd 30 sec 64>
     /pci@8,700000/fibre-channel@1/fp@0,0/ssl@w500604843e0c4bd4,7

  11. c4t500604843EC4BD4d18 <EMC-SYMMETRIX-5568 cyl 18408 alt 2 hd 15 sec 64>
     /pci@8,700000/fibre-channel@1/fp@0,0/ssl@w500604843e0c4bd4,12
```
12. c4t500604843E0C4BD4d27 <EMC-SYMMETRIX-5568 cyl 18408 alt 2 hd 15 sec 64>
   /pci@8,700000/fibre-channel@1/fp@0,0/ssd@w500604843e0c4bd4,1b
13. c4t50050763050C8786d4 <IBM-2107900-.437 cyl 4316 alt 2 hd 64 sec 256>
   /pci@8,700000/fibre-channel@1/fp@0,0/ssd@w50050763050c8786,4
14. c4t50050763050C8786d5 <IBM-2107900-.437 cyl 9205 alt 2 hd 30 sec 64>
   /pci@8,700000/fibre-channel@1/fp@0,0/ssd@w50050763050c8786,5
15. c4t50050763050C8786d6 <IBM-2107900-.437 cyl 9205 alt 2 hd 30 sec 64>
   /pci@8,700000/fibre-channel@1/fp@0,0/ssd@w50050763050c8786,6

Specify disk (enter its number):

2. Use the `df` command, as shown in Example 8-27, to determine the migration source mounted file systems. They are already under VxVM control.

**Example 8-27 Using the df command to display the currently mounted file systems**

```
# df -k /emc*
Filesystem  kbytes used   avail  capacity Mounted on
/dev/vx/dsk/dg1/vol1  8802304 5095101 3475604  60%    /emc1
/dev/vx/dsk/dg0/vol0  8802304 5349648 3236972  63%    /emc0
/dev/vx/dsk/dg2/vol2 35311616 6927899 26609769  21%    /emc2
```

3. Run the `vxprint` command to determine the relationship between the VxVM objects and associate them with the Solaris device files (Example 8-28). In the example, on disk group `dg2`, there is one disk/LUN called `dg201` with an enclosure-based name of `EMC0_2`. There is also one logical volume called `vol2`.

**Example 8-28 Using the command vxprint to show the properties of the VxVM disk groups**

```
# vxprint
```

```
Disk group: dg2
TY NAME ASSOC KSTATE LENGTH PLOFFS STATE TUTILO PUTILO
  dg dg2  dg2 - - - - - -
  dm dg201 EMCO_2 - 70624768 - - - -
  v vol2  fsgeN ENABLED 70623232 - ACTIVE - -
  pl vol2-01 vol2 ENABLED 70623232 - ACTIVE - -
  sd dg201-01 vol2-01 ENABLED 70623232 0 - - -

Disk group: dg0
TY NAME ASSOC KSTATE LENGTH PLOFFS STATE TUTILO PUTILO
  dg dg0  dg0 - - - - - -
  dm dg001 EMCO_0 - 17605888 - - - -
  v vol0  fsgeN ENABLED 17604608 - ACTIVE - -
  pl vol0-01 vol0 ENABLED 17604608 - ACTIVE - -
  sd dg001-01 vol0-01 ENABLED 17604608 0 - - -

Disk group: dg1
TY NAME ASSOC KSTATE LENGTH PLOFFS STATE TUTILO PUTILO
  dg dg1  dg1 - - - - - -
  dm dg101 EMCO_1 - 17605888 - - - -
  v vol1  fsgeN ENABLED 17604608 - ACTIVE - -
  pl vol1-01 vol1 ENABLED 17604608 - ACTIVE - -
  sd dg101-01 vol1-01 ENABLED 17604608 0 - - -
```
4. Use the enclosure-based name as input when running the `vxdisk` command, which provides the association to the Solaris device name as seen in Example 8-29. The device files are at the bottom of the output under the numpaths heading.

Example 8-29 Using the command `vxdisk list` to show the native device names

```
$ vxdisk list EMC0_0
Device:   EMC0_0
devicetag: EMC0_0
type:      auto
hostid:    SunFire280Rtic2
disk:      name=fd001 id=1174034174.32.SunFire280Rtic2
group:     name=fd0 id=1174034176.34.SunFire280Rtic2
info:      format=cddisk,privoffset=256,pubslice=2,privslice=2
flags:     online ready private autoconfig autoimport imported
pubpaths:  block=/dev/vx/dmp/EMC0_0s2 char=/dev/vx/rdmp/EMC0_0s2
guid:      {66fbe82e-1dd2-11b2-9096-0003ba17ecd4}
udid:      EMC%SYMMETRIX%F700975%F750CD00
site:      -
version:   3.1
iosize:    min=512 (bytes) max=2048 (blocks)
pub:       slice=2 offset=65792 len=17605888 disk_offset=0
private:   slice=2 offset=256 len=65536 disk_offset=0
update:    time=1174047713 seqno=0.10
ssb:       actual_seqno=0.0
headers:   0 240
configs:   count=1 len=48144
logs:      count=1 len=7296
Defined regions:
  config priv 000048-000239[000192]: copy=01 offset=000000 enabled
  config priv 000256-048207[047952]: copy=01 offset=000192 enabled
  log priv 048208-055503[007296]: copy=01 offset=000000 enabled
  lockrgn priv 055504-055647[000144]: part=00 offset=000000
Multipathing information:
numpaths:  2
c2t500604843E0C4BC8d29s2        state=enabled
c4t500604843E0C4BD4d27s2        state=enabled
```

The relationships between the Solaris device files and the VxVM disk group and disk name in the example environment is summarized in Table 8-5.

<table>
<thead>
<tr>
<th>Native device names</th>
<th>Enclosure-based name</th>
<th>Diskgroup</th>
<th>Diskname</th>
<th>Volname</th>
</tr>
</thead>
<tbody>
<tr>
<td>c2t500604843E0C4BC8d29</td>
<td>EMC0_0</td>
<td>dg0</td>
<td>dg001</td>
<td>vol0</td>
</tr>
<tr>
<td>c4t500604843E0C4BD4d27</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c2t500604843E0C4BC8d20</td>
<td>EMC0_1</td>
<td>dg1</td>
<td>dg101</td>
<td>vol1</td>
</tr>
<tr>
<td>c4t500604843E0C4BD4d18</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c2t500604843E0C4BC8d9</td>
<td>EMC0_2</td>
<td>dg1</td>
<td>dg201</td>
<td>vol2</td>
</tr>
<tr>
<td>c4t500604843E0C4BD4d7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In the example, disk numbers 3 (or 10), 4 (or 11), and 5 (or 12) will be migrated.

5. Select the disk in the menu.
6. Issue the `partition` to print the partition table.

7. Issue the `print` to print the partition information.

In the example, note that slice 2 of disk 3 is 33.71 GB. Similarly, disk 4 and 5 are 8.43 GB each. A slice in Solaris is a partition on the disk. Typically slice 2 refers to the entire disk.

Table 8-6 provides an overview of the source LUNs sizes.

<table>
<thead>
<tr>
<th>Native device names</th>
<th>Source (enclosure-based naming scheme)</th>
<th>Size of source (GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>c2l50060a4843E0C4BCBd29</td>
<td>EMC0_0</td>
<td>8.43</td>
</tr>
<tr>
<td>c4l50060a4843E0C4BD4d27</td>
<td></td>
<td></td>
</tr>
<tr>
<td>c2l50060a4843E0C4BCBd20</td>
<td>EMC0_1</td>
<td>8.43</td>
</tr>
<tr>
<td>c4l50060a4843E0C4BD4d18</td>
<td></td>
<td></td>
</tr>
<tr>
<td>c2l50060a4843E0C4BCBd9</td>
<td>EMC0_2</td>
<td>33.71</td>
</tr>
<tr>
<td>c4l50060a4843E0C4BD4d7</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Preparation of migration volumes

Prepare the target LUNs to be used as mirror targets by performing the following steps:

- Verify that the target LUNs are visible from the operating system.
- Label them if you have not already done so.

Investigating the VxVM migration target device status

In this example, three DS8000 migration destination LUNs were requested from the DS8000 storage administrator. These LUNs are slightly larger than the source LUNs: Two 8.43 GB LUN and one 33.72 GB LUN. Although the two smaller LUNs seem to have the same size as the source LUNs (8.43 GB), they differ slightly in the number of blocks. Allocation tasks and complementary tasks such as zoning have been completed successfully.

To investigate the target device status, perform the following steps:

1. Run the `vxdisk list` command to see an overview of what is currently defined to VxVM (Example 8-30). The DS8000 LUNs are currently seen by VxVM but not VxVM initialized yet (online invalid). There are gaps in the naming sequence (IBM_DS8x000_0, IBM_DS8x000_4 and IBM_DS8x000_5) because more DS8000 LUNs were mapped to this Sun server, but were later removed.

   **Example 8-30** Use vxdisk list to see what is currently under control of VxVM

   ```bash
   # vxdisk list
   DEVICE   TYPE      DISK    GROUP    STATUS          
   EMC0_0   auto:cdsdisk dg001  dg0      online          
   EMC0_1   auto:cdsdisk dg101  dg1      online          
   EMC0_2   auto:cdsdisk dg201  dg2      online          
   IBM_DS8x000_0 auto:none   -        -        online invalid
   IBM_DS8x000_4 auto:none   -        -        online invalid
   IBM_DS8x000_5 auto:none   -        -        online invalid
   c1t0d0s2  auto:none     -        -        online invalid
   c1t1d0s2  auto:none     -        -        online invalid
   #
   ```
2. Use the `vxdisk list` command to understand the relationship between the DS8000 enclosure-based names and the underlying Solaris device names as shown in Example 8-31.

**Example 8-31  Using the command vxdisk list to show the native device names**

```bash
# vxdisk list IBM_DS8x000_0
Device: IBM_DS8x000_0
devicetag: IBM_DS8x000_0
type: auto
info: format=none
flags: online ready private autoconfig invalid
pubpaths: block=/dev/vx/dmp/IBM_DS8x000_0s2 char=/dev/vx/rdmp/IBM_DS8x000_0s2
guid: -
udid: IBM%5F2107%5F75L4741%5F6005076305FFC78600000000003005
site: -
Multipathing information:
numpaths: 2
c2t50050763050C0786d5s2 state=enabled
c4t50050763050C8786d5s2 state=enabled
#
# vxdisk list IBM_DS8x000_4
Device: IBM_DS8x000_4
devicetag: IBM_DS8x000_4
type: auto
info: format=none
flags: online ready private autoconfig invalid
pubpaths: block=/dev/vx/dmp/IBM_DS8x000_4s2 char=/dev/vx/rdmp/IBM_DS8x000_4s2
guid: -
udid: IBM%5F2107%5F75L4741%5F6005076305FFC78600000000003004
site: -
Multipathing information:
numpaths: 2
c2t50050763050C0786d4s2 state=enabled
c4t50050763050C8786d4s2 state=enabled
#
# vxdisk list IBM_DS8x000_5
Device: IBM_DS8x000_5
devicetag: IBM_DS8x000_5
type: auto
info: format=none
flags: online ready private autoconfig invalid
pubpaths: block=/dev/vx/dmp/IBM_DS8x000_5s2 char=/dev/vx/rdmp/IBM_DS8x000_5s2
guid: -
udid: IBM%5F2107%5F75L4741%5F6005076305FFC78600000000003006
site: -
Multipathing information:
numpaths: 2
c2t50050763050C0786d6s2 state=enabled
c4t50050763050C8786d6s2 state=enabled
```
3. Run the `format` command to discover the size of the target LUNs. Table 8-7 summarizes the source and destination LUNs in the example. It shows the native device special file names, and the associated enclosure-based naming scheme names and the respective size.

Table 8-7  Showing the source and target volumes

<table>
<thead>
<tr>
<th>Source</th>
<th>Size of source (GB)</th>
<th>Target</th>
<th>Size of target (GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMC0_0</td>
<td>8.43</td>
<td>IBM_DS8x000_0</td>
<td>8.43</td>
</tr>
<tr>
<td></td>
<td>c21500604843E0C4B</td>
<td>c2150050763050C078</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CBd29</td>
<td>6d5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>c41500604843E0C4B</td>
<td>c4150050763050C878</td>
<td></td>
</tr>
<tr>
<td></td>
<td>D4d27</td>
<td>6d5</td>
<td></td>
</tr>
<tr>
<td>EMC0_1</td>
<td>8.43</td>
<td>IBM_DS8x000_5</td>
<td>8.43</td>
</tr>
<tr>
<td></td>
<td>c21500604843E0C4B</td>
<td>c2150050763050C078</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CBd20</td>
<td>6d6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>c41500604843E0C4B</td>
<td>c4150050763050C878</td>
<td></td>
</tr>
<tr>
<td></td>
<td>D4d18</td>
<td>6d6</td>
<td></td>
</tr>
<tr>
<td>EMC0_2</td>
<td>33.71</td>
<td>IBM_DS8x000_4</td>
<td>33.72</td>
</tr>
<tr>
<td></td>
<td>c21500604843E0C4B</td>
<td>c2150050763050C078</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CBd9</td>
<td>6d4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>c41500604843E0C4B</td>
<td>c4150050763050C878</td>
<td></td>
</tr>
<tr>
<td></td>
<td>D4d7</td>
<td>6d4</td>
<td></td>
</tr>
</tbody>
</table>

**Taking IBM DS8000 LUNs under control of VxVM**

Place the target devices under VxVM control. There is more than one way to accomplish this task. This example uses the `vxdiskadm` CLI command. During this process, you also assign the DS8000 LUNs that are about to become VxVM-initialized to the existing disk groups. The following example shows the procedural tasks required to assign enclosure IBM_DS8x000_0 to a VxVM disk group and give it a VxVM name. Although not shown, the same tasks are successfully completed on enclosures IBM_DS8x000_4 and IBM_DS8x000_5.

To bring the LUNs under VxVM control, perform these steps:

1. Use the `vxdiskadm` command to initialize the LUN in VxVM (Example 8-32).

   **Example 8-32 Using the vxdiskadm CLI command to VxVM-initialize IBM_DS8x000_0**

   Volume Manager Support Operations
   Menu: VolumeManager/Disk

   1. **Add or initialize one or more disks**
   2. Encapsulate one or more disks
   3. Remove a disk
   4. Remove a disk for replacement
   5. Replace a failed or removed disk
   6. Mirror volumes on a disk
   7. Move volumes from a disk
   8. Enable access to (import) a disk group
   9. Remove access to (deport) a disk group
   10. Enable (online) a disk device
   11. Disable (offline) a disk device
   12. Mark a disk as a spare for a disk group
   13. Turn off the spare flag on a disk
   14. Unrelocate subdisks back to a disk
   15. Exclude a disk from hot-relocation use
16. Make a disk available for hot-relocation use
17. Prevent multipathing/Suppress devices from VxVM's view
18. Allow multipathing/Unsuppress devices from VxVM's view
19. List currently suppressed/non-multipathed devices
20. Change the disk naming scheme
21. Get the newly connected/zoned disks in VxVM view
22. Change/Display the default disk layouts
23. Mark a disk as allocator-reserved for a disk group
24. Turn off the allocator-reserved flag on a disk
25. List disk information
26. Display help about menu
27. Display help about the menuing system
28. Exit from menus

Select an operation to perform: 1

2. Select option 1 as shown in Example 8-33.

Example 8-33 Having selected option 1, add or initialize one or more disks

Add or initialize disks
Menu: VolumeManager/Disk/AddDisks
Use this operation to add one or more disks to a disk group. You can add the selected disks to an existing disk group or to a new disk group that will be created as a part of the operation. The selected disks may also be added to a disk group as spares. Or they may be added as nohotuses to be excluded from hot-relocation use. The selected disks may also be initialized without adding them to a disk group leaving the disks available for use as replacement disks.

More than one disk or pattern may be entered at the prompt. Here are some disk selection examples:

all: all disks
c3 c4t2: all disks on both controller 3 and controller 4, target 2
c3t4d2: a single disk (in the c#t#d# naming scheme)
xyz_0 : a single disk (in the enclosure based naming scheme)
xyz_ : all disks on the enclosure whose name is xyz

Select disk devices to add: [pattern-list],all,list,q,?] IBM_DS8x000_0
Here is the disk selected. Output format: [Device_Name]

IBM_DS8x000_0

Continue operation? [y,n,q,?] (default: y) y
You can choose to add this disk to an existing disk group, a new disk group, or leave the disk available for use by future add or replacement operations. To create a new disk group, select a disk group name that does not yet exist. To leave the disk available for future use, specify a disk group name of "none".

Which disk group [group,none,list,q,?] (default: dg0)
Use a default disk name for the disk? [y,n,q,?] (default: y)
Add disk as a spare disk for dg0? [y,n,q,?] (default: n)
Exclude disk from hot-relocation use? [y,n,q,?] (default: n)

3. Enter the disk names to be initialized as shown in Example 8-34.

Example 8-34  Having selected option 1, add or initialize one or more disks (cont'd)

Add site tag to disk? [y,n,q,?] (default: n)
The selected disks will be added to the disk group dg0 with
default disk names.
IBM_DS8x000_0

Continue with operation? [y,n,q,?] (default: y)
The following disk device has a valid VTOC, but does not appear to have
been initialized for the Volume Manager. If there is data on the disk
that should NOT be destroyed you should encapsulate the existing disk
partitions as volumes instead of adding the disk as a new disk.
Output format: [Device_Name]

IBM_DS8x000_0
Encapsulate this device? [y,n,q,?] (default: y) n
IBM_DS8x000_0

Instead of encapsulating, initialize? [y,n,q,?] (default: n) y
Initializing device IBM_DS8x000_0.

Enter desired private region length
[<privlen>,q,?] (default: 65536)
VxVM NOTICE V-5-2-88
Adding disk device IBM_DS8x000_0 to disk group dg0 with disk
name dg002.
Add or initialize other disks? [y,n,q,?] (default: n)

4. Use the list option on the vxdiskadm CLI command verify that the DS8000 migration
target volumes are under VxVM control as shown in Example 8-35.

Example 8-35  List all devices after they are VxVM-initialized

List disk information
Menu: VolumeManager/Disk/ListDisk
Use this menu operation to display a list of disks. You can
also choose to list detailed information about the disk at
a specific disk device address.

Enter disk device or "all" [<address>,all,q,?] (default: all)

<table>
<thead>
<tr>
<th>DEVICE</th>
<th>DISK</th>
<th>GROUP</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMC0_0</td>
<td>dg001</td>
<td>dg0</td>
<td>online</td>
</tr>
<tr>
<td>EMC0_1</td>
<td>dg101</td>
<td>dg1</td>
<td>online</td>
</tr>
<tr>
<td>EMC0_2</td>
<td>dg201</td>
<td>dg2</td>
<td>online</td>
</tr>
<tr>
<td>IBM_DS8x000_0</td>
<td>dg002</td>
<td>dg0</td>
<td>online</td>
</tr>
<tr>
<td>IBM_DS8x000_4</td>
<td>dg202</td>
<td>dg2</td>
<td>online</td>
</tr>
</tbody>
</table>
You can also check using the `vxdisk list` and `vxprint` commands as shown in Example 8-36.

**Example 8-36  VxVM verification commands**

```plaintext
# vxdisk list

<table>
<thead>
<tr>
<th>DEVICE</th>
<th>TYPE</th>
<th>DISK</th>
<th>GROUP</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMC0_0</td>
<td>auto:cdsdisk</td>
<td>dg001</td>
<td>dg0</td>
<td>online</td>
</tr>
<tr>
<td>EMC0_1</td>
<td>auto:cdsdisk</td>
<td>dg101</td>
<td>dg1</td>
<td>online</td>
</tr>
<tr>
<td>EMC0_2</td>
<td>auto:cdsdisk</td>
<td>dg201</td>
<td>dg2</td>
<td>online</td>
</tr>
<tr>
<td>IBM_DS8x000_0</td>
<td>auto:cdsdisk</td>
<td>dg002</td>
<td>dg0</td>
<td>online</td>
</tr>
<tr>
<td>IBM_DS8x000_4</td>
<td>auto:cdsdisk</td>
<td>dg202</td>
<td>dg2</td>
<td>online</td>
</tr>
<tr>
<td>IBM_DS8x000_5</td>
<td>auto:cdsdisk</td>
<td>dg102</td>
<td>dg1</td>
<td>online</td>
</tr>
<tr>
<td>c1t0d0s2</td>
<td>auto:none</td>
<td>-</td>
<td>-</td>
<td>online invalid</td>
</tr>
<tr>
<td>c1t1d0s2</td>
<td>auto:none</td>
<td>-</td>
<td>-</td>
<td>online invalid</td>
</tr>
</tbody>
</table>

# vxprint

Disk group: dg2

```plaintext

```plaintext
<table>
<thead>
<tr>
<th>TY</th>
<th>NAME</th>
<th>ASSOC</th>
<th>KSTATE</th>
<th>LENGTH</th>
<th>PLOFFS</th>
<th>STATE</th>
<th>TUTIL0</th>
<th>PUTIL0</th>
</tr>
</thead>
<tbody>
<tr>
<td>dg</td>
<td>dg2</td>
<td>dg2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

```

```plaintext
| dm | dg201 | EMCO_2      | -      | 70624768 | -      | -     | -      | -      |

```

```plaintext
| dm | dg202 | IBM_DS8x000_4 | -      | 70647552 | -      | -     | -      | -      |

```

```plaintext
| v  | vol12 | fsgen      | ENABLED | 70623232 | -      | ACTIVE | -      | -      |

```

```plaintext
| pl | vol12-01 | vol12 | ENABLED | 70623232 | -      | ACTIVE | -      | -      |

```

```plaintext
| sd | dg201-01 | vol2-01 | ENABLED | 70623232 | 0      | -      | -      | -      |

```

Disk group: dg0

```plaintext

```plaintext
<table>
<thead>
<tr>
<th>TY</th>
<th>NAME</th>
<th>ASSOC</th>
<th>KSTATE</th>
<th>LENGTH</th>
<th>PLOFFS</th>
<th>STATE</th>
<th>TUTIL0</th>
<th>PUTIL0</th>
</tr>
</thead>
<tbody>
<tr>
<td>dg</td>
<td>dg0</td>
<td>dg0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

```

```plaintext
| dm | dg001 | EMCO_0      | -      | 17605888 | -      | -     | -      | -      |

```

```plaintext
| dm | dg002 | IBM_DS8x000_0 | -      | 17607808 | -      | -     | -      | -      |

```

```plaintext
| v  | vol10 | fsgen      | ENABLED | 17604608 | -      | ACTIVE | -      | -      |

```

```plaintext
| pl | vol10-01 | vol10 | ENABLED | 17604608 | -      | ACTIVE | -      | -      |

```

```plaintext
| sd | dg001-01 | vol0-01 | ENABLED | 17604608 | 0      | -      | -      | -      |

```

Disk group: dg1

```plaintext

```plaintext
<table>
<thead>
<tr>
<th>TY</th>
<th>NAME</th>
<th>ASSOC</th>
<th>KSTATE</th>
<th>LENGTH</th>
<th>PLOFFS</th>
<th>STATE</th>
<th>TUTIL0</th>
<th>PUTIL0</th>
</tr>
</thead>
<tbody>
<tr>
<td>dg</td>
<td>dg1</td>
<td>dg1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

```

```plaintext
| dm | dg101 | EMCO_1      | -      | 17605888 | -      | -     | -      | -      |

```

```plaintext
| dm | dg102 | IBM_DS8x000_5 | -      | 17607808 | -      | -     | -      | -      |

```

```plaintext
| v  | vol11 | fsgen      | ENABLED | 17604608 | -      | ACTIVE | -      | -      |

```
Establishing the mirrors
To establish the VxVM mirrors, perform the following steps:

1. Run the `vxdiskadm` CLI command and select option 6, Mirror volumes on a disk as shown in Example 8-37. This example refers to dg0 and vol0.

   **Example 8-37 Establishing mirroring**
   Select an operation to perform: 6

   Mirror volumes on a disk
   Menu: VolumeManager/Disk/Mirror
   This operation can be used to mirror volumes on a disk. These volumes can be be mirrored onto another disk or onto any available disk space. Volumes will not be mirrored if they are already mirrored. Also, volumes that are comprised of more than one subdisk will not be mirrored.

   Mirroring volumes from the boot disk will produce a disk that can be used as an alternate boot disk.

   At the prompt below, supply the name of the disk containing the volumes to be mirrored.

   Enter disk name [<disk>,list,q,?] list
   Disk group: dg2

<table>
<thead>
<tr>
<th>DM NAME</th>
<th>DEVICE</th>
<th>TYPE</th>
<th>PRIVLEN</th>
<th>PUBLEN</th>
<th>STATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>dm dg201</td>
<td>EMC0_2</td>
<td>auto</td>
<td>65536</td>
<td>70624768</td>
<td>-</td>
</tr>
<tr>
<td>dm dg202</td>
<td>IBM_DS8x000_4 auto</td>
<td>65536</td>
<td>70647552</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

   Disk group: dg0

<table>
<thead>
<tr>
<th>DM NAME</th>
<th>DEVICE</th>
<th>TYPE</th>
<th>PRIVLEN</th>
<th>PUBLEN</th>
<th>STATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>dm dg001</td>
<td>EMC0_0</td>
<td>auto</td>
<td>65536</td>
<td>17605888</td>
<td>-</td>
</tr>
<tr>
<td>dm dg002</td>
<td>IBM_DS8x000_0 auto</td>
<td>65536</td>
<td>17607808</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

   Disk group: dg1

<table>
<thead>
<tr>
<th>DM NAME</th>
<th>DEVICE</th>
<th>TYPE</th>
<th>PRIVLEN</th>
<th>PUBLEN</th>
<th>STATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>dm dg101</td>
<td>EMC0_1</td>
<td>auto</td>
<td>65536</td>
<td>17605888</td>
<td>-</td>
</tr>
<tr>
<td>dm dg102</td>
<td>IBM_DS8x000_5 auto</td>
<td>65536</td>
<td>17607808</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

   Enter disk name [<disk>,list,q,?] dg001
   You can choose to mirror volumes from disk dg001 onto any available disk space, or you can choose to mirror onto a specific disk. To mirror to a specific disk, select the name of that disk. To mirror to any available disk space, select "any".

   Enter destination disk [<disk>,list,q,?] (default: any) dg002
The requested operation is to mirror all volumes on disk dg001 in disk group dg0 onto available disk space on disk dg002.

VxVM NOTICE V-5-2-3650 This operation can take a long time to complete.

Continue with operation? [y,n,q,?] (default: y)
VxVM vxmirror INFO V-5-2-22  Mirror volume vol0 ... 

2. Repeat step 1 for all other disks.
3. Use the vxtask list command to monitor the progress of the mirroring activity (Example 8-38).

Example 8-38  Using vxtask list to display the mirroring progress

```# vxtask list
TASKID  PTID  TYPE/STATE  PCT  PROGRESS
170     ATCOPY/R 13.58%  0/17604608/2390016  PLXATT vol1 vol1-02 dg1
#```

4. The mirrors are in sync when the vxtask list command no longer shows any ongoing activity (Example 8-39).
5. Enter vxprint and make sure that new plexes and subdisks have been created in each disk group.

Example 8-39  Using vxtask list and then vxprint

```# vxtask list
TASKID  PTID  TYPE/STATE  PCT  PROGRESS
# vxprint
Disk group: dg2
TY  NAME    ASSOC    KSTATE    LENGTH   PLOFFS   STATE    TUTIL0  PUTIL0
dg  dg2      dg2       -         -         -         -         -         -
dm dg201    EMCO_2    -         70624768 -         -         -         -
dm dg202    IBM_DS8x000_4 -       70647552 -         -         -         -
v  vol2     fsgen     ENABLED  70623232 -         ACTIVE   -         -
pl vol2-01  vol2     ENABLED  70623232 -         ACTIVE   -         -
sd dg201-01 vol2-01  ENABLED  70623232 0        ACTIVE   -         -
pl vol2-02  vol2     ENABLED  70623232 -         ACTIVE   -         -
sd dg202-01 vol2-02  ENABLED  70623232 0        ACTIVE   -         -

Disk group: dg0
TY  NAME    ASSOC    KSTATE    LENGTH   PLOFFS   STATE    TUTIL0  PUTIL0
dg  dg0      dg0       -         -         -         -         -         -
dm dg001    EMCO_0    -         17605888 -         -         -         -
dm dg002    IBM_DS8x000_0 -       17607808 -         -         -         -
v  vol0     fsgen     ENABLED  17604608 -         ACTIVE   -         -
pl vol0-01  vol0     ENABLED  17604608 -         ACTIVE   -         -
sd dg001-01 vol0-01  ENABLED  17604608 0        ACTIVE   -         -
pl vol0-02  vol0     ENABLED  17604608 -         ACTIVE   -         -
sd dg002-01 vol0-02  ENABLED  17604608 0        ACTIVE   -         -```
Removing the source LUNs from the mirrors

Remove the source LUNs from the mirrors using the `vxplex` command (Example 8-40). In the example, a subsequent `vxprint` shows the results of the `vxplex` operation against `dg0`. The source EMC LUN has been removed from the mirror but is still part of the disk group.

**Example 8-40   Using vxplex to get the EMC LUN out of the mirror**

```
# vxplex -g dg0 -o rm dis vol0-01
# vxprint -g dg0
```

<table>
<thead>
<tr>
<th>TY</th>
<th>NAME</th>
<th>ASSOC</th>
<th>KSTATE</th>
<th>LENGTH</th>
<th>PLOFFS</th>
<th>STATE</th>
<th>TUTIL0</th>
<th>PUTIL0</th>
</tr>
</thead>
<tbody>
<tr>
<td>dg</td>
<td>dg0</td>
<td>dg0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>dm</td>
<td>dg001</td>
<td>EMC0_0</td>
<td>-</td>
<td>17605888</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>dm</td>
<td>dg002</td>
<td>IBM_DS8x000_0</td>
<td>-</td>
<td>17607808</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>v</td>
<td>vol0</td>
<td>fsgen</td>
<td>ENABLED</td>
<td>17604608</td>
<td>-</td>
<td>ACTIVE</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>pl</td>
<td>vol0-02</td>
<td>vol0</td>
<td>ENABLED</td>
<td>17604608</td>
<td>-</td>
<td>ACTIVE</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>sd</td>
<td>dg002-01</td>
<td>vol0-02</td>
<td>ENABLED</td>
<td>17604608</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

The same task needs to be done for `dg1` and `dg2`, as shown in Example 8-41.

**Example 8-41   Using vxplex to get the EMC LUN out of the mirrors**

```
# vxplex -g dg1 -o rm dis vol1-01
# vxplex -g dg2 -o rm dis vol2-01
# vxprint
```

<table>
<thead>
<tr>
<th>TY</th>
<th>NAME</th>
<th>ASSOC</th>
<th>KSTATE</th>
<th>LENGTH</th>
<th>PLOFFS</th>
<th>STATE</th>
<th>TUTIL0</th>
<th>PUTIL0</th>
</tr>
</thead>
<tbody>
<tr>
<td>dg</td>
<td>dg2</td>
<td>dg2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>dm</td>
<td>dg201</td>
<td>EMC0_2</td>
<td>-</td>
<td>70624768</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>dm</td>
<td>dg202</td>
<td>IBM_DS8x000_4</td>
<td>-</td>
<td>70647552</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>v</td>
<td>vol12</td>
<td>fsgen</td>
<td>ENABLED</td>
<td>70623232</td>
<td>-</td>
<td>ACTIVE</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>pl</td>
<td>vol12-02</td>
<td>vol12</td>
<td>ENABLED</td>
<td>70623232</td>
<td>-</td>
<td>ACTIVE</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>sd</td>
<td>dg202-01</td>
<td>vol12-02</td>
<td>ENABLED</td>
<td>70623232</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Chapter 8. Using mirroring techniques

Removing the source LUNs from the VxVM disk groups

Remove the source EMC LUNs from the disk group by entering `vxdg -g <dgN> rmdisk ...`, 0 <= N <= 2 as seen in Example 8-42. After the process is complete, display the disk groups again to verify that the EMC LUNs are removed.

Example 8-42   Removing EMC LUNs from the disk groups and displaying the results

```
# vxdg -g dg0 rmdisk dg001
# vxdg -g dg1 rmdisk dg101
# vxdg -g dg2 rmdisk dg201
# vxdisk list

DEVICE       TYPE            DISK         GROUP        STATUS
---           ----            ---            ---          ----
EMC0_0       auto:cdsdisk    -            -            online
EMC0_1       auto:cdsdisk    -            -            online
EMC0_2       auto:cdsdisk    -            -            online
IBM_DS8x000_0 auto:cdsdisk    dg002        dg0          online
IBM_DS8x000_4 auto:cdsdisk    dg202        dg2          online
IBM_DS8x000_5 auto:cdsdisk    dg102        dg1          online
c1t0d0s2     auto:none       -            -            online invalid
clt1d0s2     auto:none       -            -            online invalid
```

Verifying the migration

Verify that the migration is complete by entering `vxprint` (Example 8-43).

Example 8-43   Entering vxprint to see that EMC LUNs are not contained in any of the disk groups

```
# vxprint

Disk group: dg2

TY NAME         ASSOC        KSTATE   LENGTH   PLOFFS   STATE    TUTIL0  PUTIL0
---             ---           ---       ---        ---       ---        ---        ---
dg dg2          dg2           -         -         -         -         -         -
dm dg202        IBM_DS8x000_4 - 70647552 -         -         -         -
```
<table>
<thead>
<tr>
<th>v</th>
<th>vol2</th>
<th>fsgen</th>
<th>ENABLED</th>
<th>70623232</th>
<th>ACTIVE</th>
<th>-</th>
<th>-</th>
</tr>
</thead>
<tbody>
<tr>
<td>pl</td>
<td>vol2-02</td>
<td>vol2</td>
<td>ENABLED</td>
<td>70623232</td>
<td>ACTIVE</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>sd</td>
<td>dg202-01</td>
<td>vol2-02</td>
<td>ENABLED</td>
<td>70623232 0</td>
<td>ACTIVE</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Disk group: dg0

<table>
<thead>
<tr>
<th>TY</th>
<th>NAME</th>
<th>ASSOC</th>
<th>KSTATE</th>
<th>LENGTH</th>
<th>PLOFFS</th>
<th>STATE</th>
<th>T_UTIL0</th>
<th>P_UTIL0</th>
</tr>
</thead>
<tbody>
<tr>
<td>dg</td>
<td>dg0</td>
<td>dg0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>dm</td>
<td>dg002</td>
<td>IBM_DS8x000_0</td>
<td>-</td>
<td>17607808</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>v</th>
<th>vol0</th>
<th>fsgen</th>
<th>ENABLED</th>
<th>17604608</th>
<th>ACTIVE</th>
<th>-</th>
<th>-</th>
</tr>
</thead>
<tbody>
<tr>
<td>pl</td>
<td>vol0-02</td>
<td>vol0</td>
<td>ENABLED</td>
<td>17604608</td>
<td>ACTIVE</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>sd</td>
<td>dg002-01</td>
<td>vol0-02</td>
<td>ENABLED</td>
<td>17604608 0</td>
<td>ACTIVE</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Disk group: dg1

<table>
<thead>
<tr>
<th>TY</th>
<th>NAME</th>
<th>ASSOC</th>
<th>KSTATE</th>
<th>LENGTH</th>
<th>PLOFFS</th>
<th>STATE</th>
<th>T_UTIL0</th>
<th>P_UTIL0</th>
</tr>
</thead>
<tbody>
<tr>
<td>dg</td>
<td>dg1</td>
<td>dg1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>dm</td>
<td>dg102</td>
<td>IBM_DS8x000_5</td>
<td>-</td>
<td>17607808</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

The migration is now complete and the data is located exclusively on DS8000 LUNs. There might be some additional tasks left beyond the data migration itself, such as disconnecting the EMC LUNs from the system.

### 8.6 Data migration using HP-UX Volume Manager mirroring

This section addresses the tasks required to migrate existing EMC LUNs to new DS8000 LUNs, using the HP-UX Volume Manager mirroring capabilities. The process addressed here is fairly generic. The number of LUNs and the type of storage subsystem they originate from do not really matter: The process is the same for one or multiple LUNs from any subsystem. Depending on the specifics of your environment, the process can be disruptive to applications, requiring reboots and file system unmounts.

In the scenario that follows, the volume to be migrated is part of a volume group (VG). In this example, the EMC storage LUN to be migrated is integrated into a VG called vg_emc_to_ibm.
8.6.1 High-level plan for migration using HP_UX Volume Manager mirroring

The high-level steps are shown in Table 8-8.

<table>
<thead>
<tr>
<th>Process</th>
<th>Commands</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1.</strong> Get an overview of the attached</td>
<td>ioscan -fnC disk</td>
<td>Discover the DS8000 LUNs on the HP-UX host system.</td>
</tr>
<tr>
<td>disks.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2.</strong> Create the device special files.</td>
<td>insf -e</td>
<td>You must create the device special files before you can work with the DS8000 LUNs.</td>
</tr>
<tr>
<td><strong>Step 3.</strong> Verify that the device special</td>
<td>ioscan -fnC disk</td>
<td>Rescan.</td>
</tr>
<tr>
<td>files were created.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4.</strong> Prepare the target volumes to</td>
<td>1. pvcreate /dev/rdsk/c125t0d0</td>
<td>Use one LUN as an argument for pvcreate to make the new device ready to be used in a volume group (VG).</td>
</tr>
<tr>
<td>be included into a volume group.</td>
<td>2. vgdisplay -v /dev/vg_emc_to_ibm</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5.</strong> Integrate the target volumes</td>
<td>vgextend /dev/vg_emc_to_ibm /dev/dsk/c126t0d0</td>
<td>Bring an IBM storage LUN into VG vg_emc_to_ibm.</td>
</tr>
<tr>
<td>into the respective volume group.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 6.</strong> Assign an alternative path for</td>
<td>vgextend /dev/vg_emc_to_ibm /dev/dsk/c125t0d0</td>
<td>For redundancy, provide an alternative path for the DS8000 LUN.</td>
</tr>
<tr>
<td>the LUN that just became integrated.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 7.</strong> Set up a mirror.</td>
<td>lvextend -m 1 /dev/vg_emc_to_ibm/lvol1 /dev/dsk/c125t0d0</td>
<td>Create a mirror for the logical volume and force that mirror to be on /dev/dsk/c125t0d0 (one of the paths to the IBM LUN).</td>
</tr>
<tr>
<td><strong>Step 8.</strong> Remove the source LUN.</td>
<td>1. vgreduce /dev/vg_emc_to_ibm /dev/dsk/c123t1d4</td>
<td>Remove the alternative link to the EMC device.</td>
</tr>
<tr>
<td></td>
<td>2. lvreduce -m 0 /dev/vg_emc_to_ibm/lvol1 /dev/dsk/c121t1d6</td>
<td>Reduce the number of mirror copies.</td>
</tr>
<tr>
<td></td>
<td>3. vgreduce /dev/vg_emc_to_ibm /dev/dsk/c121t1d6</td>
<td>Remove the last remaining path to the EMC LUN.</td>
</tr>
<tr>
<td><strong>Step 9.</strong> Verify that the mirror is</td>
<td>vgdisplay -v /dev/vg_emc_to_ibm</td>
<td>Verify that no EMC device special file is being referred to.</td>
</tr>
<tr>
<td>removed.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

8.6.2 Detailed steps for migration using HP-UUX Volume Manager mirroring

The migration procedure has the following detailed steps:

- Getting an overview of the attached disks
- Creating device special files
- Preparing the target volumes for inclusion in a volume group
- Integrating the target volumes into the respective volume group
- Assigning an alternative path to the LUN
- Setting up a mirror
- Removing the source LUN
- Verifying the migration
Getting an overview of the attached disks
Check which disks in your environment are visible by the HP-UX host using the `ioscan -fnC` disk command. In the example environment shown in Example 8-44, apart from the EMC LUNs, the HP-UX host sees four DS8000 LUNs. You see twice the number of LUNs than are available because the storage administrator assigned two LUNs that are presented through two ESS ports.

Example 8-44  Using ioscan to show the LUNs that are visible to the OS

```
# ioscan -fnC disk
Class     I  H/W Path       Driver   S/W State   H/W Type    Description
==========================================================================
disk      0  0/0/1/1.2.0    sdisk    CLAIMED     DEVICE       FUJITSU MAJ3182MC
          /dev/dsk/c1t2d0     /dev/rdsk/c1t2d0
disk      1  0/0/2/1.2.0    sdisk    CLAIMED     DEVICE       HP          DVD-ROM 305
          /dev/dsk/c3t2d0     /dev/rdsk/c3t2d0
disk    266  0/4/0/0.18.50.0.36.0.0  sdisk    CLAIMED     DEVICE       IBM          2107900
          /dev/dsk/c123t0d0   /dev/rdsk/c123t0d0
disk    264  0/4/0/0.18.51.0.36.0.0  sdisk    CLAIMED     DEVICE       IBM          2107900
          /dev/dsk/c123t1d4   /dev/rdsk/c123t1d4
disk    262  0/4/0/0.18.60.0.0.0.0.0   sdisk    CLAIMED     DEVICE       EMC          SYMMETRIX
          /dev/dsk/c121t0d0   /dev/rdsk/c121t0d0
disk    263  0/4/0/0.18.60.0.0.1.4   sdisk    CLAIMED     DEVICE       EMC          SYMMETRIX
          /dev/dsk/c121t1d6   /dev/rdsk/c121t1d6
disk    265  0/7/0/0.18.50.0.36.0.0  sdisk    CLAIMED     DEVICE       IBM          2107900
          /dev/dsk/c123t0d0   /dev/rdsk/c123t0d0
disk    264  0/7/0/0.18.51.0.36.0.0  sdisk    CLAIMED     DEVICE       IBM          2107900
          /dev/dsk/c123t1d4   /dev/rdsk/c123t1d4
disk    266  0/7/0/0.18.61.0.0.0.0.0   sdisk    CLAIMED     DEVICE       EMC          SYMMETRIX
          /dev/dsk/c121t0d0   /dev/rdsk/c121t0d0
disk    261  0/7/0/0.18.61.0.0.1.6   sdisk    CLAIMED     DEVICE       EMC          SYMMETRIX
          /dev/dsk/c121t1d6   /dev/rdsk/c121t1d6
```

Creating device special files
You must create the device special files for the DS8000 LUNs before you can work with those LUNs. To create the files, perform the following steps:

1. Run the `insf -e` command as shown in Example 8-45.

Example 8-45  Using insf -e to create the device special files

```
# insf -e
insf: Installing special files for btlan instance 0 address 0/0/0/0
insf: Installing special files for sctl instance 0 address 0/0/1/0.7.0
insf: Installing special files for sdisk instance 0 address 0/0/1/1.2.0
insf: Installing special files for sctl instance 1 address 0/0/1/1.7.0
insf: Installing special files for sctl instance 2 address 0/0/2/0.7.0
insf: Installing special files for sdisk instance 1 address 0/0/2/1.2.0
insf: Installing special files for sctl instance 3 address 0/0/2/1.7.0
insf: Installing special files for asio0 instance 0 address 0/0/4/1
insf: Installing special files for hub instance 0 address 0/2/0/0.1
insf: Installing special files for btlan instance 1 address 0/3/0/0.4/0
insf: Installing special files for btlan instance 2 address 0/3/0/0.5/0
insf: Installing special files for btlan instance 3 address 0/3/0/0.6/0
insf: Installing special files for btlan instance 4 address 0/3/0/0.7/0
insf: Installing special files for sdisk instance 266 address 0/4/0/0.18.50.0.36.0.0
insf: Installing special files for sctl instance 23 address 0/4/0/0.18.50.255.0.0
insf: Installing special files for sdisk instance 264 address 0/4/0/0.18.51.0.36.0.0
insf: Installing special files for sctl instance 21 address 0/4/0/0.18.51.255.0
```
insf: Installing special files for sdisk instance 262 address 0/4/0/18.60.0.0.
insf: Installing special files for sdisk instance 263 address 0/4/0/18.60.0.0.
insf: Installing special files for sctl instance 20 address 0/4/0/18.60.255.0.
insf: Installing special files for sdisk instance 265 address 0/7/0/18.50.0.36
insf: Installing special files for sctl instance 22 address 0/7/0/18.50.255.0.
insf: Installing special files for sdisk instance 267 address 0/7/0/18.51.0.36
insf: Installing special files for sctl instance 19 address 0/7/0/18.61.255.0.
insf: Installing special files for pseudo driver cn
insf: Installing special files for pseudo driver mm
insf: Installing special files for pseudo driver devkrs
insf: Installing special files for pseudo driver ptym
insf: Installing special files for pseudo driver ptys
insf: Installing special files for pseudo driver ip
insf: Installing special files for pseudo driver arp
insf: Installing special files for pseudo driver rawip
insf: Installing special files for pseudo driver tcp
insf: Installing special files for pseudo driver udp
insf: Installing special files for pseudo driver stcpmap
insf: Installing special files for pseudo driver nuls
insf: Installing special files for pseudo driver netqa
insf: Installing special files for pseudo driver dmem
insf: Installing special files for pseudo driver diag0
insf: Installing special files for pseudo driver telm
insf: Installing special files for pseudo driver tels
insf: Installing special files for pseudo driver tlclts
insf: Installing special files for pseudo driver tlcots
insf: Installing special files for pseudo driver iomem
insf: Installing special files for pseudo driver tlcontsod
insf: Installing special files for pseudo driver dmp
insf: Installing special files for pseudo driver vols
insf: Installing special files for pseudo driver dev_config
insf: Installing special files for pseudo driver strlog
insf: Installing special files for pseudo driver sad
insf: Installing special files for pseudo driver echo
insf: Installing special files for pseudo driver dlpi
insf: Installing special files for pseudo driver ptm
insf: Installing special files for pseudo driver pts
insf: Installing special files for pseudo driver beep
insf: Installing special files for pseudo driver framebuffer
insf: Installing special files for pseudo driver diag1
insf: Installing special files for pseudo driver klog
2. Enter the `ioscan -fnC disk` command again to verify that the device special files were created (Example 8-46).

**Example 8-46 Using iosscan -fnC disk**

```
# ioscan -fnC disk
```

<table>
<thead>
<tr>
<th>Class</th>
<th>I</th>
<th>H/W Path</th>
<th>Driver</th>
<th>S/W State</th>
<th>H/W Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>0/0/1.2.0</td>
<td>sdisk</td>
<td>CLAIMED</td>
<td>DEVICE</td>
<td>FUJITSU MAJ3182MC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>/dev/dsk/c1t2d0</td>
<td></td>
<td></td>
<td></td>
<td>/dev/rdsk/c1t2d0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0/0/2.2.0</td>
<td>sdisk</td>
<td>CLAIMED</td>
<td>DEVICE</td>
<td>HP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>/dev/dsk/c3t2d0</td>
<td></td>
<td></td>
<td></td>
<td>DVD-ROM 305</td>
</tr>
<tr>
<td>disk</td>
<td>266</td>
<td>0/4/0.18.50.0.36.0.0</td>
<td>sdisk</td>
<td>CLAIMED</td>
<td>DEVICE</td>
<td>IBM 2107900</td>
</tr>
<tr>
<td></td>
<td></td>
<td>/dev/dsk/c127t0d0</td>
<td></td>
<td></td>
<td></td>
<td>/dev/rdsk/c127t0d0</td>
</tr>
<tr>
<td></td>
<td>264</td>
<td>0/4/0.18.51.0.36.0.0</td>
<td>sdisk</td>
<td>CLAIMED</td>
<td>DEVICE</td>
<td>IBM 2107900</td>
</tr>
<tr>
<td></td>
<td></td>
<td>/dev/dsk/c125t0d0</td>
<td></td>
<td></td>
<td></td>
<td>/dev/rdsk/c125t0d0</td>
</tr>
<tr>
<td></td>
<td>262</td>
<td>0/4/0.18.60.0.0.0.0.0</td>
<td>sdisk</td>
<td>CLAIMED</td>
<td>DEVICE</td>
<td>EMC SYMMETRIX</td>
</tr>
<tr>
<td></td>
<td></td>
<td>/dev/dsk/c123t0d0</td>
<td></td>
<td></td>
<td></td>
<td>/dev/rdsk/c123t0d0</td>
</tr>
<tr>
<td></td>
<td>263</td>
<td>0/4/0.18.60.0.0.1.4</td>
<td>sdisk</td>
<td>CLAIMED</td>
<td>DEVICE</td>
<td>EMC SYMMETRIX</td>
</tr>
<tr>
<td></td>
<td></td>
<td>/dev/dsk/c123t1d4</td>
<td></td>
<td></td>
<td></td>
<td>/dev/rdsk/c123t1d4</td>
</tr>
<tr>
<td></td>
<td>265</td>
<td>0/7/0.18.50.0.36.0.0</td>
<td>sdisk</td>
<td>CLAIMED</td>
<td>DEVICE</td>
<td>IBM 2107900</td>
</tr>
<tr>
<td></td>
<td></td>
<td>/dev/dsk/c126t0d0</td>
<td></td>
<td></td>
<td></td>
<td>/dev/rdsk/c126t0d0</td>
</tr>
<tr>
<td></td>
<td>267</td>
<td>0/7/0.18.51.0.36.0.0</td>
<td>sdisk</td>
<td>CLAIMED</td>
<td>DEVICE</td>
<td>IBM 2107900</td>
</tr>
<tr>
<td></td>
<td></td>
<td>/dev/dsk/c128t0d0</td>
<td></td>
<td></td>
<td></td>
<td>/dev/rdsk/c128t0d0</td>
</tr>
<tr>
<td></td>
<td>260</td>
<td>0/7/0.18.61.0.0.0.0.0</td>
<td>sdisk</td>
<td>CLAIMED</td>
<td>DEVICE</td>
<td>EMC SYMMETRIX</td>
</tr>
<tr>
<td></td>
<td></td>
<td>/dev/dsk/c121t0d0</td>
<td></td>
<td></td>
<td></td>
<td>/dev/rdsk/c121t0d0</td>
</tr>
<tr>
<td></td>
<td>261</td>
<td>0/7/0.18.61.0.0.1.6</td>
<td>sdisk</td>
<td>CLAIMED</td>
<td>DEVICE</td>
<td>EMC SYMMETRIX</td>
</tr>
<tr>
<td></td>
<td></td>
<td>/dev/dsk/c121t1d6</td>
<td></td>
<td></td>
<td></td>
<td>/dev/rdsk/c121t1d6</td>
</tr>
</tbody>
</table>

--- Volume groups ---

**Preparing the target volumes for inclusion in a volume group**

After the device special files are available, use one of them as argument for the `pvcreate` command as shown in Example 8-47. This command makes the new device ready to be used in a volume group (VG).

**Example 8-47 Using pvcreate to make a LUN ready to be integrated into a volume group**

```
# pvcreate /dev/rdsk/c125t0d0
Physical volume "/dev/rdsk/c125t0d0" has been successfully created
```

Issue the command `vgdisplay -v /dev/vg_emc_to_ibm` to see the following characteristics of this particular VG:

- Size
- Physical volumes (PVs)
- Physical elements
- Logical volumes (LVs).

Example 8-48 shows the output of the `vgdisplay -v /dev/vg_emc_to_ibm` command.

**Example 8-48 Issuing vgdisplay -v /dev/vg_emc_to_ibm**

```
# vgdisplay -v /dev/vg_emc_to_ibm
```

--- Volume groups ---
### Integrating the target volumes into the respective volume group

Use the `vgextend` command to bring a DS8000 target LUN into the VG `vg_emc_to_ibm` as shown in Example 8-49.

**Example 8-49  Using the command vgextend to add an IBM LUN into the existing VG**

```bash
# vgextend /dev/vg_emc_to_ibm /dev/dsk/c126t0d0
gextend: Warning: Max_PE_per_PV for the volume group (2157) too small for this PV (2303).
  Using only 2157 PEs from this physical volume.
Current path "/dev/dsk/c123t1d4" is an alternate link, skip.
Volume group "/dev/vg_emc_to_ibm" has been successfully extended.
Volume Group configuration for /dev/vg_emc_to_ibm has been saved in /etc/lvmconf/vg_emc_to_ibm.conf
# vgdisplay
```

--- Volume groups ---

<table>
<thead>
<tr>
<th>VG Name</th>
<th>/dev/vg00</th>
</tr>
</thead>
<tbody>
<tr>
<td>VG Write Access</td>
<td>read/write</td>
</tr>
<tr>
<td>VG Status</td>
<td>available</td>
</tr>
</tbody>
</table>

--- Logical volumes ---

<table>
<thead>
<tr>
<th>LV Name</th>
<th>/dev/vg_emc_to_ibm/lvol1</th>
</tr>
</thead>
<tbody>
<tr>
<td>LV Status</td>
<td>available/syncd</td>
</tr>
<tr>
<td>LV Size (Mbytes)</td>
<td>8624</td>
</tr>
<tr>
<td>Current LE</td>
<td>2156</td>
</tr>
<tr>
<td>Allocated PE</td>
<td>2156</td>
</tr>
<tr>
<td>Used PV</td>
<td>1</td>
</tr>
</tbody>
</table>

--- Physical volumes ---

<table>
<thead>
<tr>
<th>PV Name</th>
<th>/dev/dsk/c121t1d6</th>
</tr>
</thead>
<tbody>
<tr>
<td>PV Name</td>
<td>/dev/dsk/c123t1d4 Alternate Link</td>
</tr>
<tr>
<td>PV Status</td>
<td>available</td>
</tr>
<tr>
<td>Total PE</td>
<td>2156</td>
</tr>
<tr>
<td>Free PE</td>
<td>0</td>
</tr>
<tr>
<td>Autoswitch</td>
<td>On</td>
</tr>
</tbody>
</table>
The warning message "vgextend: Warning: Max_PE_per_PV for the volume group (2157) too small for this PV (2303)" is displayed because the DS8000 LUN size exceeds the size of the EMC LUN. When creating a volume group, the number of physical elements is set to a default value. If the number of physical elements calculated when creating the volume group exceeds that default value, the calculated number is used.

To avoid this problem, perform the following steps:

- Make sure that the administrator defines enough physical elements. Check the elements using option -e when running the `vgcreate` command.
- Create the target LUNs so that their number of physical elements do not exceed the number of physical elements that is valid for this volume group. In this case, the target source LUNs would need to the same size as the source LUNs.

You can also accept that some space on the target volume is wasted.

**Tip:** More recent versions of HP-UX (HP-UX 11i v3 and later) allow you to modify the physical element number dynamically by issuing the `vgmodify` command.
Assigning an alternative path to the LUN

For redundancy, provide and verify an alternative path for the DS8000 LUN as shown in Example 8-50.

Example 8-50  Provide an alternative path for the target volume and verify successful completion

```bash
# vgextend /dev/vg_emc_to_ibm /dev/dsk/c125t0d0
vgextend: Warning: Max_PE_per_PV for the volume group (2157) too small for this PV (2303).
        Using only 2157 PEs from this physical volume.
Current path "/dev/dsk/c123t1d4" is an alternate link, skip.
Volume group "/dev/vg_emc_to_ibm" has been successfully extended.
Volume Group configuration for "/dev/vg_emc_to_ibm" has been saved in /etc/lvmconf
    /vg_emc_to_ibm.conf
# vgdisplay -v /dev/vg_emc_to_ibm
--- Volume groups ---
VG Name                     /dev/vg_emc_to_ibm
VG Write Access             read/write
VG Status                   available
Max LV                      255
Cur LV                      1
Open LV                     1
Max PV                      16
Cur PV                      2
Act PV                      2
Max PE per PV               2157
VGDA                        4
PE Size (Mbytes)            4
Total PE                    4313
Alloc PE                    2156
Free PE                     2157
Total PVG                   0
Total Spare PVs             0
Total Spare PVs in use      0
--- Logical volumes ---
LV Name                     /dev/vg_emc_to_ibm/lvol1
LV Status                   available/syncd
LV Size (Mbytes)            8624
Current LE                  2156
Allocated PE                2156
Used PV                     1
--- Physical volumes ---
PV Name                     /dev/dsk/c121t1d6
PV Name                     /dev/dsk/c123t1d4
    Alternate Link
PV Status                   available
Total PE                    2156
Free PE                     0
Autoswitch                  On
PV Name                     /dev/dsk/c126t0d0
PV Name                     /dev/dsk/c125t0d0
    Alternate Link
PV Status                   available
Total PE                    2157
Free PE                     2157
Autoswitch                  On
```
In the example, /dev/dsk/c125t0d0 is labeled Alternate Link. It is labeled this way because /dev/dsk/c126t0d0 and /dev/dsk/c125t0d0 are two paths to the same volume rather than two separate volumes.

**Setting up a mirror**

Use the `lvextend -m 1` command to create a mirror for the logical volume. Force that mirror to be on /dev/dsk/c125t0d0 (one of the paths to the IBM LUN) as shown in Example 8-51. Wait until the mirror becomes established.

*Example 8-51   Mirror to the target device*

```bash
# lvextend -m 1 /dev/vg_emc_to_ibm/lvol1 /dev/dsk/c125t0d0
Device file path "/dev/dsk/c125t0d0" is an alternate path to the Physical Volume. Using Primary Link "/dev/dsk/c126t0d0".
The newly allocated mirrors are now being synchronized. This operation will take some time. Please wait ....
Logical volume "/dev/vg_emc_to_ibm/lvol1" has been successfully extended.
Volume Group configuration for /dev/vg_emc_to_ibm has been saved in /etc/lvmconf/vg_emc_to_ibm.conf
```

**Removing the source LUN**

The following steps address how to remove the source (EMC) LUN:

1. Remove the alternate link to the EMC device using the `vgreduce` command as shown in Example 8-52.

*Example 8-52   Using the command vgreduce to remove one path to the EMC LUN*

```bash
# vgreduce /dev/vg_emc_to_ibm /dev/dsk/c123t1d4
Device file path "/dev/dsk/c123t1d4" is an alternate path.
Volume group "/dev/vg_emc_to_ibm" has been successfully reduced.
Volume Group configuration for /dev/vg_emc_to_ibm has been saved in /etc/lvmconf/vg_emc_to_ibm.conf
```

2. Use the `lvreduce` command to reduce the number of mirror copies so that the EMC LUN is not part of the mirror (Example 8-53).

*Example 8-53   Using the command lvreduce to remove the EMC LUN from the mirror*

```bash
# lvreduce -m 0 /dev/vg_emc_to_ibm/lvol1 /dev/dsk/c121t1d6
Logical volume "/dev/vg_emc_to_ibm/lvol1" has been successfully reduced.
Volume Group configuration for /dev/vg_emc_to_ibm has been saved in /etc/lvmconf/vg_emc_to_ibm.conf
```

3. Remove the last remaining path to the EMC LUN with the `vgreduce` command as shown in Example 8-54.

*Example 8-54   Remove the last remaining path to the EMC LUN*

```bash
vgreduce /dev/vg_emc_to_ibm /dev/dsk/c121t1d6
Volume group "/dev/vg_emc_to_ibm" has been successfully reduced.
Volume Group configuration for /dev/vg_emc_to_ibm has been saved in /etc/lvmconf/vg_emc_to_ibm.conf
#
```
Verifying the migration

Verify that no EMC device special file is displayed by issuing the `vgdisplay` command (Example 8-55).

**Example 8-55  Using vgdisplay to display VG components**

```bash
# vgdisplay -v /dev/vg_emc_to_ibm
--- Volume groups ---
VG Name                     /dev/vg_emc_to_ibm
VG Write Access             read/write
VG Status                   available
Max LV                      255
Cur LV                      1
Open LV                     1
Max PV                      16
Cur PV                      1
Act PV                      1
Max PE per PV               2157
VGDA                        2
PE Size (Mbytes)            4
Total PE                    2157
Alloc PE                    2156
Free PE                     1
Total PVG                   0
Total Spare PVs             0
Total Spare PVs in use      0
--- Logical volumes ---
LV Name                     /dev/vg_emc_to_ibm/lvol1
LV Status                   available/syncd
LV Size (Mbytes)            8624
Current LE                  2156
Allocated PE                2156
Used PV                     1
--- Physical volumes ---
PV Name                     /dev/dsk/c126t0d0
PV Name                     /dev/dsk/c125t0d0        Alternate Link
PV Status                   available
Total PE                    2157
Free PE                     1
Autoswitch                  On
```

### 8.7 Data migration using AIX LVM mirroring

This section addresses mirroring using the AIX Logical Volume Manager (LVM) to migrate data from an ESS to a DS8000.

There are two possible methods to migrate using AIX LVM mirroring:

- Using `migratepv -l`
- Using `mklvcopy` and `syncvg`
You can connect the replacement DS8000 in parallel with the existing ESS. Assign the LUNs from the DS8000 to the same compatible host server HBAs that the ESS LUNs are assigned to. In this way, you can mirror LUNs or volumes on the existing unit to the replacement unit using software (LVM) techniques.

After the volumes are mirrored, break the mirror from the existing unit and remove the old unit. Prepare the DS8000 with the correct Array to LUN spread and LUN size considerations as it applies to the current setup of the ESS. For more information about DS8000 preparation, see 2.4, “Preparing DS8000 for data migration” on page 27.

### 8.7.1 ESS to DS8000 migration under AIX LVM test environment

The physical test environment shown in Figure 8-1 on page 269 is composed of the following components:

- IBM eServer™ pSeries® 615 Model 6C3 (now discontinued), 7029-6C3 AIX level 5.3.0.0, maintenance level 04
  - IBM SDD PCM for AIX V53 version 2.1.2.0 Disk Device driver
  - Emulex HBAs running firmware version 9.1.2.14, Feature Code 5704, Part Number 00P4295
- ESS model 800 Storage running microcode level 2.4.4.45
- DS8100 model 921 running firmware level SEA 5.2.400.437, bundle 6.2.400.76
- Brocade Switch type 44, Fabric OS firmware version 5.1.0b

The example environment is three existing ESS data volumes (left side of the diagram in solid lines) configured on the Windows server and running live applications. Introduce the DS8000 into the environment and migrate the 2105 volumes (source volumes) to the DS8000 volumes (target volumes). The three DS8000 volumes were created of equal or greater capacity using type ds as in the DS8000. The target volume capacities (sizes, are shown on the right side of the diagram in dotted lines.)
Chapter 8. Using mirroring techniques

8.7.2 High-level migration plan using the AIX LVM migratepv -l

The high-level steps for a migration based on the `migratepv -l` command are shown in Table 8-9. The table illustrates an ESS to DS8000 migration example. You can apply the same process to other open system storage platforms.

Table 8-9   Example migration procedure using migratepv -l

<table>
<thead>
<tr>
<th>Process</th>
<th>Commands</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1.</strong> Identify the ESS source LUNs.</td>
<td><code>lsvpcfg</code></td>
<td>Gather information about the AIX host, such as the number of ESS LUNs and sizes.</td>
</tr>
<tr>
<td><strong>Step 2.</strong> Assign the DS8000 LUNs to the host.</td>
<td><code>chvolgrp -dev image_id -action add-volume vol_# V3</code></td>
<td>On the DS8000, use the DS8000 CLI (dscli) to assign the LUNs. The <code>image_id</code> is the DS8000 ID and <code>vol_#</code> is the number of the DS8000 volume in hex.</td>
</tr>
<tr>
<td><strong>Step 3.</strong> Discover the DS8000 LUNs.</td>
<td><code>cfgmgr</code></td>
<td>On the AIX host, discover the newly assigned LUNs.</td>
</tr>
<tr>
<td><strong>Step 4.</strong> Determine the difference between the ESS and DS8000 LUNs.</td>
<td><code>lsvpcfg</code></td>
<td>This command works using the SDD driver on the AIX host. The source LUNs are ESS and the target LUNs are DS8000. Look at the serial numbers of the LUNs in the output.</td>
</tr>
</tbody>
</table>
### Process | Commands | Explanation
--- | --- | ---
**Step 5.** Identify the sizes of the DS8000 target LUNs. | `bootinfo -s vpath#` | Where `#` is the number of the vpath.
**Step 6.** Move the DS8000 LUNs into the appropriate VGs. | `extendvg vg-name vpath#` | Where `vg_name` is the name of the VG and `#` is the number of the vpath.
**Step 7.** Verify that the DS8000 LUNs are added to the VG. | `lsvg -p vg_name` | Where `vg_name` is the name of the VG.
**Step 8.** Identify the logical volumes (LVs) to migrate. | `lsvg -l vg_name` | Where `vg_name` is the name of the VG.
**Step 9.** Copy LV data from the ESS source LUNs to the DS8000 target LUNs. | `migratepv -l lv_name` | Where `lv_name` is the name of the LV.
**Step 10.** Verify that the LUNs are copied. | `lsvg -p vg_name` `lspv -l vpath#` | Where `vg_name` is the name of the VG and `#` is the number of the vpath.
**Step 11.** Remove the ESS source LUNs from the VGs and verify that the source ESS LUNs are removed from the VG. | `reducevg vg_name vpath#` `lsvg -p vg_name` | Where `vg_name` is the name of the VG and `#` is the number of the vpath.
**Step 12.** Delete the device definitions from the host ODM. | `rmdev -dl vpath#` `rmdev -dl hdisk#` | Where `#` is the number of the vpath and appropriate hdisk.
**Step 13.** Verify that the device definitions are removed | `lsdev -Cc disk` | Check that there are no defined disks.
**Step 14.** Remove the source zone definition from the switch fabric. | n/a | Removes data path between AIX host and disk subsystem.
**Step 15.** In the ESS, unassign the LUNs from the host server. | 1. Click Modify Volume Assignments. 2. In the Volume assignments window, click First from the Host Nicknames list. 3. Click Perform Sort. 4. Look for the first assigned volume assigned to the host. 5. Press the Ctrl key and click the volume. 6. In the Action list, select Unassign selected volume(s) to target hosts. 7. In the Target Hosts list, select all the targeted volumes using the Ctrl key. 8. Click Perform Configuration Update. 9. Click OK. | This sequence is done from the ESS Specialist GUI.
8.7.3 Detailed steps using migratepv -l

The migration procedure has the following detailed steps, using AIX level 5.2 with both the 2105 and the 2107 SDD drivers:

- Identifying the ESS source LUNs in the targeted VGs
- Assigning the DS8000 LUNs to the AIX host
- Discovering the DS8000 target LUNs
- Determining the differences between the ESS and DS8000 LUNs
- Identifying the sizes of the DS8000 LUNs
- Moving the DS8000 LUNs into the VGs appropriately
- Verifying that the DS8000 LUNs are added to the VG
- Identifying the LVs to migrate
- Copying LV data from the source LUNs to the DS8000 target LUNs
- Verifying that the LUNs are copied
- Removing the ESS source LUNs from the VGs
- Deleting the device definitions from the host ODM
- Verifying that the device definitions are removed
- Removing the source zone definition
- Unassigning the source LUNs from the host server

Identifying the ESS source LUNs in the targeted VGs

By issuing `lsvg -p` against the target `vg_name`, you can determine the size of the LUNs that you need to assign on the DS8000. See Example 8-56.

Multiply the number of physical partitions (PPs) for the vpath by the PP size of the volume group. This calculation gives you the LUN size in MB. The formula is:

\[(\text{LUNPPs}) * \text{vgPPsize} = \text{LUN size}\]

If you take the Total PPs for vpath2 and multiply it by the vg PP size of 64, you get 4736 MB:

\[(74) * (64) = 4736 \text{ MB}\]

**Example 8-56**  Results from running the lsvg -p command

```
# lsvg -p chuckvg
chuckvg:
PV_NAME           PV STATE          TOTAL PPs   FREE PPs    FREE DISTRIBUTION
vpath2            active            74          0           00..00..00..00..00
vpath3            active            74          1           00..00..00..00..01
vpath4            active            14          1           00..00..00..00..01
```

Assigning the DS8000 LUNs to the AIX host

The example scenario has the volumes and volgrp already created. Assign the new volumes 4004 - 4006 to V3, and initialize the host connections to the DS8000 using the `dscli` command as shown in Example 8-57.

**Example 8-57**  The dscli command to assign storage to the AIX host

```
mkfbvol -dev IBM.2107-75L4741 -extpool P4 -name 8300_#h -type ds -cap 5 4004-4005
mkfbvol -dev IBM.2107-75L4741 -extpool P4 -name 8300_#h -type ds -cap 1 4006
mkvolgrp -dev IBM.2107-75L4741 IBM.2107-7581981 scsimask -volume 4004-4006 V3
mkhostconnect -dev IBM.2107-75L4741 -wwname 10000000C93E007C -hosttype pSeries -volgrp V5 p615-tic-6_A0
mkhostconnect -dev IBM.2107-75L4741 -wwname 10000000C93E0059 -hosttype pSeries -volgrp V5 p615-tic-6_A1
```
To verify that LUNs are assigned to V3, issue the `showvolgrp` command as shown in Example 8-58.

**Example 8-58  Displaying LUNs in a volgrp using the dscli**

```bash
dscli> showvolgrp -dev IBM.2107-75L4741 V3
Date/Time: March 28, 2007 12:56:45 PM CST IBM DSCLI Version: 5.2.400.304 DS: IBM.2107-75L4741
Name p615_tic6 vg
ID V3
Type SCSI Mask
Vols 4004 4005 4006
```

LUNs 4004-4005 are assigned to volgrp V3.

**Discovering the DS8000 target LUNs**

Assign the DS8000 LUNs and run `cfgmgr` on the AIX host system to discover the newly assigned LUNs.

**Determining the differences between the ESS and DS8000 LUNs**

Identify the DS8000 and ESS LUNs by the LUN ID displayed in the output of the `lsvpcfg` command as shown in Example 8-59.

**Example 8-59  Identifying ESS and DS8000 LUNs**

```bash
# lsvpcfg
vpath2 (Avail pv chuckvg) 70022665 = hdisk19 (Avail ) hdisk22 (Avail ) hdisk25 (Avail ) hdisk34 (Avail )
vpath3 (Avail pv chuckvg) 70122665 = hdisk20 (Avail ) hdisk23 (Avail ) hdisk26 (Avail ) hdisk35 (Avail )
vpath4 (Avail pv chuckvg) 70422665 = hdisk21 (Avail ) hdisk24 (Avail ) hdisk27 (Avail ) hdisk36 (Avail )
vpath5 (Avail ) 75L47414005 = hdisk14 (Avail ) hdisk17 (Avail ) hdisk29 (Avail ) hdisk32 (Avail )
vpath6 (Avail ) 75L47414006 = hdisk15 (Avail ) hdisk18 (Avail ) hdisk30 (Avail ) hdisk33 (Avail )
vpath7 (Avail ) 75L47414004 = hdisk16 (Avail ) hdisk28 (Avail ) hdisk37 (Avail )
```

The ESS LUN IDs consist of the first three digits, which giving the LUN hex ID, and the last five digits that are the ESS serial number. The DS8000 LUNs are the opposite way: The DS8000 serial number is the first eight digits, and the last four digits are the LUN hex ID. For example, `vpath 5` is a DS8000 LUN, so `75L4741` is the serial number of the unit and `4005` is the LUN ID. In the LUN ID, `40` is the LSS and `05` is the LUN ID in hex.

**Identifying the sizes of the DS8000 LUNs**

You need to know the sizes of the DS8000 LUNs to move them into the VGs. Remember that the DS8000 target LUN should be at least the same size or larger than the source ESS LUN when using the `migratepv` command. If you choose to use the `mklvcopy` command, check that there is enough space in the VG on the DS8000 LUNs collectively to create the mirrors. Issue the `bootinfo -s vpath/hdisk` command to identify the sizes in MB (Example 8-60).

**Example 8-60  Output showing the results of the bootinfo command**

```bash
# bootinfo -s vpath5
5120
```
Moving the DS8000 LUNs into the VGs appropriately

Issue the `extendvg` command to move the DS8000 LUNs into the targeted volume groups as shown in Example 8-61.

**Example 8-61 Using the extendvg command to move LUNs into a VG**

```
extendvg chuckvg vpath5 vpath6 vpath7
0516-1254 extendvg:Changing the PVID in the ODM
```

Verifying that the DS8000 LUNs are added to the VG

To verify that the vpaths are added to the VG correctly, run the `lsvg -p` command as shown in Example 8-62.

**Example 8-62 Output results from using the lsvg -p command**

```
# lsvg -p chuckvg
chuckvg:

<table>
<thead>
<tr>
<th>PV_NAME</th>
<th>PV_STATE</th>
<th>TOTAL PPs</th>
<th>FREE PPs</th>
<th>FREE DISTRIBUTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>vpath2</td>
<td>active</td>
<td>74</td>
<td>0</td>
<td>00..00..00..00..00</td>
</tr>
<tr>
<td>vpath3</td>
<td>active</td>
<td>74</td>
<td>1</td>
<td>00..00..00..00..01</td>
</tr>
<tr>
<td>vpath4</td>
<td>active</td>
<td>14</td>
<td>1</td>
<td>00..00..00..00..01</td>
</tr>
<tr>
<td>vpath5</td>
<td>active</td>
<td>79</td>
<td>79</td>
<td>16..16..15..16..16</td>
</tr>
<tr>
<td>vpath6</td>
<td>active</td>
<td>15</td>
<td>15</td>
<td>03..03..03..03..03</td>
</tr>
<tr>
<td>vpath7</td>
<td>active</td>
<td>79</td>
<td>79</td>
<td>16..16..15..16..16</td>
</tr>
</tbody>
</table>
```

Identifying the LVs to migrate

To identify the LVs to migrate from the source ESS LUNs to the target DS8000 LUNs, issue the command `lsvg -l` as shown in Example 8-63.

**Example 8-63 The output results by issuing the lsvg -l command against a VG**

```
# lsvg -l chuckvg
chuckvg:

<table>
<thead>
<tr>
<th>LV_NAME</th>
<th>TYPE</th>
<th>LPs</th>
<th>PPs</th>
<th>PVs</th>
<th>LV_STATE</th>
<th>MOUNT POINT</th>
</tr>
</thead>
<tbody>
<tr>
<td>ess5GB1v1</td>
<td>jfs2</td>
<td>73</td>
<td>73</td>
<td>1</td>
<td>open/syncd</td>
<td>/ess5GB1</td>
</tr>
<tr>
<td>ess5GB1v2</td>
<td>jfs2</td>
<td>73</td>
<td>73</td>
<td>1</td>
<td>open/syncd</td>
<td>/ess5GB2</td>
</tr>
<tr>
<td>ess1GB1v</td>
<td>jfs2</td>
<td>13</td>
<td>13</td>
<td>1</td>
<td>open/syncd</td>
<td>/ess1GB</td>
</tr>
<tr>
<td>loglv00</td>
<td>jfs2log</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>open/syncd</td>
<td>N/A</td>
</tr>
</tbody>
</table>
```
Copying LV data from the source LUNs to the DS8000 target LUNs

Use the `migratepv` command to accomplish the migration. To move physical partitions from `vpath2` to `vpath5`, issue the `migratepv -l` command against the VG as shown in Example 8-64.

**Note:** Do not use the `migratepv -p` command. An error can leave the logical volume with data on both the target and source LUNs in an unknown state.

**Example 8-64** Using the `migratepv -l` command to migrate data

```
# migratepv -l ess5GBlv1 vpath2 vpath5
```

Verifying that the LUNs are copied

When the migration is completed, verify that the data contents on `vpath2` are migrated to the target DS8000 LUN (`vpath5`). You can check by issuing the `lsvg -p` command as shown in Example 8-65.

**Example 8-65** Results from issuing the `lsvg -p` command

```
# lsvg -p chuckvg
chuckvg:
   PV_NAME           PV STATE          TOTAL PPs   FREE PPs    FREE DISTRIBUTION
   vpath2            active            74          73          15..15..14..15..14
   vpath3            active            74          1           00..00..00..00..01
   vpath4            active            14          1           00..00..00..00..01
   vpath5            active            79          6           00..00..00..00..06
   vpath6            active            15          15          03..03..03..03..03
   vpath7            active            79          79          16..16..16..16..16
```

Notice that `vpath2` now has 73 PPs free and `vpath5` has the contents on it.

Another way to verify that the data is by issuing the `lspv -l` command against the source and target vpaths as shown in Example 8-66.

**Example 8-66** Results from issuing the `lspv -l` command against the vpath

```
# lspv -l vpath5
vpath5:
   LV NAME               LPs   PPs   DISTRIBUTION          MOUNT POINT
   ess5GBlv1             73    73    16..16..16..16..10    /ess5GB1

# lspv -l vpath2
vpath2:
   LV NAME               LPs   PPs   DISTRIBUTION          MOUNT POINT
   loglv00               1     1     00..00..00..00..01    N/A
```

Notice that there is still one PP on `vpath2` that needs to be moved.
Move loglv00 from the targeted ESS LUN vpath2 to the targeted DS8000 LUN by following the process shown in Example 8-67.

**Example 8-67  Using the migratepv -l command and the output results**

<table>
<thead>
<tr>
<th>LV NAME</th>
<th>LPs</th>
<th>PPs</th>
<th>DISTRIBUTION</th>
<th>MOUNT POINT</th>
</tr>
</thead>
<tbody>
<tr>
<td>ess5GBlv1</td>
<td>73</td>
<td>73</td>
<td>16..16..15..16..10</td>
<td>/ess5GBlv1</td>
</tr>
<tr>
<td>loglv00</td>
<td>1</td>
<td>1</td>
<td>00..00..00..00..01</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Removing the ESS source LUNs from the VGs

Remove vpath2 from the VG by issuing the `reducevg` command. When the process is complete, verify that the source ESS LUN is removed from the VG by using the `lsvg -p` command (Example 8-68).

**Example 8-68  Removing and verifying the LUN from the VG**

<table>
<thead>
<tr>
<th>PV_NAME</th>
<th>PV STATE</th>
<th>TOTAL PPs</th>
<th>FREE PPs</th>
<th>FREE DISTRIBUTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>vpath3</td>
<td>active</td>
<td>74</td>
<td>1</td>
<td>00..00..00..00..01</td>
</tr>
<tr>
<td>vpath4</td>
<td>active</td>
<td>14</td>
<td>1</td>
<td>00..00..00..00..01</td>
</tr>
<tr>
<td>vpath5</td>
<td>active</td>
<td>79</td>
<td>5</td>
<td>00..00..00..00..05</td>
</tr>
<tr>
<td>vpath6</td>
<td>active</td>
<td>15</td>
<td>15</td>
<td>03..03..03..03..03</td>
</tr>
<tr>
<td>vpath7</td>
<td>active</td>
<td>79</td>
<td>79</td>
<td>16..16..15..16..16</td>
</tr>
</tbody>
</table>

Check the vpaths and associated hdisks by running the `lsvpcfg` command as shown in Example 8-69.

**Example 8-69  Results of running the lsvpcfg command**

<table>
<thead>
<tr>
<th>PV_NAME</th>
<th>PV STATE</th>
<th>TOTAL PPs</th>
<th>FREE PPs</th>
<th>FREE DISTRIBUTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>vpath2</td>
<td>(Avail pv) 70022665 = hdisk25 (Avail ) hdisk2 (Avail ) hdisk3 (Avail ) hdisk4 (Avail )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vpath3</td>
<td>(Avail pv) 70122665 = hdisk20 (Avail ) hdisk23 (Avail ) hdisk26 (Avail ) hdisk35 (Avail )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vpath4</td>
<td>(Avail pv) 70422665 = hdisk21 (Avail ) hdisk24 (Avail ) hdisk27 (Avail ) hdisk36 (Avail )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vpath5</td>
<td>(Avail pv chuckvg) 75L47414005 = hdisk14 (Avail ) hdisk17 (Avail ) hdisk29 (Avail ) hdisk32 (Avail )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vpath6</td>
<td>(Avail pv chuckvg) 75L47414006 = hdisk15 (Avail ) hdisk18 (Avail ) hdisk30 (Avail ) hdisk33 (Avail )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vpath7</td>
<td>(Avail pv chuckvg) 75L47414004 = hdisk16 (Avail ) hdisk28 (Avail ) hdisk37 (Avail ) hdisk38 (Avail )</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Deleting the device definitions from the host ODM

Remove the device definitions for all the source ESS vpaths and all their associated hdisk paths. For example, to remove vpath2 and its associated hdisk paths, issue the `rmdev` commands as shown in Example 8-70.

**Example 8-70 Deleting the device definitions**

```
rmdev -dl vpath2
rmdev -dl hdisk19
rmdev -dl hdisk22
rmdev -dl hdisk25
```

In Example 8-70, note that only vpath2 and its associated disks were deleted. Use the same process to delete all the vpaths and the associated hdisks that are owned by vpath3 and vpath4.

Verifying that the device definitions are removed

Verify that there are no defined disks any more by issuing the `lsdev -Cc disk` command as shown in Example 8-71.

**Example 8-71 Results from the lsdev -Cc disk command**

```
# lsdev -Cc disk
hdisk0  Available 1S-08-00-4,0 16 Bit LVD SCSI Disk Drive
hdisk1  Available 1S-08-00-5,0 16 Bit LVD SCSI Disk Drive
hdisk14 Available 1V-08-01     IBM FC 2107
hdisk15 Available 1V-08-01     IBM FC 2107
hdisk16 Available 1H-08-01     IBM FC 2107
hdisk17 Available 1H-08-01     IBM FC 2107
hdisk18 Available 1H-08-01     IBM FC 2107
hdisk28 Available 1H-08-01     IBM FC 2107
hdisk29 Available 1H-08-01     IBM FC 2107
hdisk30 Available 1H-08-01     IBM FC 2107
hdisk32 Available 1V-08-01     IBM FC 2107
hdisk33 Available 1V-08-01     IBM FC 2107
hdisk37 Available 1V-08-01     IBM FC 2107
hdisk38 Available 1V-08-01     IBM FC 2107
vpath5  Available              Data Path Optimizer Pseudo Device Driver
vpath6  Available              Data Path Optimizer Pseudo Device Driver
vpath7  Available              Data Path Optimizer Pseudo Device Driver
```

Removing the source zone definition

Remove the source zone definition from the switch fabric.
Unassigning the source LUNs from the host server
Unassign the source ESS volumes from the storage unit and verify that the ESS volumes are no longer showing on the system. See Figure 8-25.

![Modify Volume Assignments](image)

**Figure 8-25 Unassigning ESS LUNs**

### 8.7.4 High-level migration plan using AIX LVM mklvcopy and syncvg commands

The AIX LVM is flexible. Under certain constraints or requirements for data spreading or consolidation across LUNs, it might be easier to use the `mklvcopy` and `syncvg` commands. If you choose to use the `mklvcopy` command, check that there is enough space in the VG on the DS8000 LUNs collectively to create the mirrors. Table 8-10 shows the sequence, commands, and a brief explanation of this scenario.

**Table 8-10 Example migration procedure using mklvcopy**

<table>
<thead>
<tr>
<th>Process</th>
<th>Commands</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1.</strong> Identify the ESS source LUNs.</td>
<td><code>lsvpcfg</code></td>
<td>Gather information about the AIX host, such as the number of ESS LUNs and sizes.</td>
</tr>
<tr>
<td><strong>Step 2.</strong> Assign the DS8000 LUNs to the host.</td>
<td><code>chvolgrp -dev image_id -action add-volume vol_# V3</code></td>
<td>On the DS8000, use the DS8000 CLI (dscli) to assign the LUNs. Where <code>image_id</code> is the DS8000 ID and <code>vol_#</code> is the number of the DS8000 volume in hex.</td>
</tr>
<tr>
<td><strong>Step 3.</strong> Discover the DS8000 LUNs.</td>
<td><code>cfgmgr</code></td>
<td>On the AIX host, discover the newly assigned LUNs.</td>
</tr>
<tr>
<td>Process</td>
<td>Commands</td>
<td>Explanation</td>
</tr>
<tr>
<td>---------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>Step 4. Determine the difference between the ESS and DS8000 LUNs.</td>
<td>lsvpcfg</td>
<td>This command works using the SDD driver on the AIX host. The source LUNs are ESS and the target LUNs are DS8000. Look at the serial numbers of the LUNs in the output.</td>
</tr>
<tr>
<td>Step 5. Identify the sizes of the DS8000 target LUNs.</td>
<td>bootinfo -s vpath#</td>
<td>Where # is the number of the vpath.</td>
</tr>
<tr>
<td>Step 6. Move the DS8000 LUNs into the VGs appropriately.</td>
<td>extendvg vg-name vpath#</td>
<td>Where vg_name is the name of the VG and # is the number of the vpath.</td>
</tr>
<tr>
<td>Step 7. Verify that the DS8000 LUNs are added to the VG.</td>
<td>lsvg -p vg_name</td>
<td>Where vg_name is the name of the VG.</td>
</tr>
<tr>
<td>Step 8. Determine how the LVs are spread across the vpaths.</td>
<td>lslv -l lv_name</td>
<td>Where lv_name is the name of the LV.</td>
</tr>
<tr>
<td>Step 9. Reserve free space on each LUN for an even spread of the data across LUNs.</td>
<td>mklv -y lvdummy vg_name PPs vpath#</td>
<td>Where vg_name is the name of the VG and # is the number of the vpath.</td>
</tr>
<tr>
<td>Step 10. Copy LV data from the ESS source LUNs to the DS8000 target LUNs.</td>
<td>mkllvcopy lv_name 2 vpath# vpath#</td>
<td>Where lv_name is the name of the LV and # is the number of the vpath of the target DS8000 LUN.</td>
</tr>
<tr>
<td>Step 11. Verify that the LV copies are correct.</td>
<td>lslv -l lv_name</td>
<td>Where lv_name is the name of the LV.</td>
</tr>
<tr>
<td>Step 12. Synchronize the LV data from the ESS source LUNs to the DS8000 target LUNs.</td>
<td>syncvg -v vg_name</td>
<td>Where vg_name is the name of the VG.</td>
</tr>
<tr>
<td>Step 13. Verify that the sync is displayed as sync’d rather than stale.</td>
<td>lsvg -l vg_name</td>
<td>If the lv still shows stale, you need to resync it before proceeding.</td>
</tr>
<tr>
<td>Step 14. Verify the source and target LUNs for each LV.</td>
<td>lslv -l lv_name</td>
<td>Where lv_name is the name of the LV.</td>
</tr>
<tr>
<td>Step 15. Remove the source copy of the LV from the ESS LUNs.</td>
<td>rmllvcopy lv_name 1 vpath#</td>
<td>Where lv_name is the name of the LV and # is the number of the vpath of the target DS8000 LUN.</td>
</tr>
<tr>
<td>Step 16. Verify that all the source ESS LUNs are free with no data.</td>
<td>lsvg -p vg_name</td>
<td>Where vg_name is the name of the VG.</td>
</tr>
<tr>
<td>Step 17. Remove the ESS source LUNs from the VGs and verify that the source ESS LUNs are removed from the VG.</td>
<td>reducevg vg_name vpath# lsvg -p vg_name</td>
<td>Where vg_name is the name of the VG and # is the number of the vpath.</td>
</tr>
<tr>
<td>Step 18. Delete the device definitions from the host ODM.</td>
<td>rmdev -dl vpath# rmdev -dl hdisk#</td>
<td>Where # is the number of the vpath and appropriate hdisk.</td>
</tr>
</tbody>
</table>
**8.7.5 Detailed migration steps using mklvcopy and syncvg**

The migration procedure has the following detailed steps, using an AIX V5.2 with both the 2105 and the 2107 SDD drivers:

- Identifying the source ESS LUNs in the targeted VGs
- Assigning the DS8000 LUNs to the AIX host
- Assigning and discover the DS8000 LUNs
- Determining the differences between the ESS and DS8000 LUN
- Identifying the sizes of the DS8000 LUNs
- Moving the DS8000 LUNs into the VGs
- Verifying that the vpaths are correctly added to the VG
- Determining how the LVs are spread across the vpaths
- Reserving free space for an even spread of the data
- Copying LV data from source LUNs to target LUNs
- Verifying that the copies are correct
- Synchronizing the copies
- Checking for stale partitions
- Verifying LV copies
- Removing the source copies from the ESS LUNs
- Verifying that all the source ESS LUNs are free with no data
- Removing the ESS source LUNs from the VGs
- Deleting the device definitions from the host ODM
- Verifying that the device definitions are removed
- Removing the source zone definition
- Unassigning the LUNs from the host server in the ESS

<table>
<thead>
<tr>
<th>Process</th>
<th>Commands</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 19. Verify that the device definitions are removed.</td>
<td>lsdev -Cc disk</td>
<td>Check that there are no defined disks.</td>
</tr>
<tr>
<td>Step 20. Remove the source zone definition from the switch fabric.</td>
<td>n/a</td>
<td>Removes data path between AIX host and disk subsystem</td>
</tr>
<tr>
<td>Step 21. In the ESS, unassign the LUNs from the host server.</td>
<td>1. Click <strong>Modify Volume Assignments</strong>. 2. In the Volume assignments window, select <strong>First</strong> in the Host Nicknames list. 3. Click <strong>Perform Sort</strong>. 4. Press the Ctrl key and select the first volume assigned to the host. 5. In the Action list, select <strong>Unassign selected volume(s) to target hosts</strong>. 6. In the Target Hosts list, select all targeted volumes using the Ctrl key. 7. Click <strong>Perform Configuration Update</strong>. 8. Click OK.</td>
<td>This sequence is done from the ESS Specialist GUI.</td>
</tr>
</tbody>
</table>
Identifying the source ESS LUNs in the targeted VGs

Issue the `lsvg -p` command against the target `vg_name` to determine the size of the LUNs that you need to assign from the DS8000 (Example 8-72). To identify the sizes of the LUNs in the targeted VG, multiply the number of PPs for the vpath by the PP size of the VG. This calculation gives you the LUN size in MB. The formula is:

\[(\text{LUNPPs}) \times \text{vgPPsize} = \text{LUN size}\]

If you take the Total PPs for `vpath2` and multiply it by the VG PP size of 64 you get 4736 MB:

\[(74) \times (64) = 4736 \text{ MB}\]

**Example 8-72**  Results from running the `lsvg -p` command

```
# lsvg -p chuckvg
chuckvg:
PV_NAME           PV STATE          TOTAL PPs   FREE PPs    FREE DISTRIBUTION
vpath2            active            74          0           00..00..00..00..00
vpath3            active            74          1           00..00..00..00..01
vpath4            active            14          1           00..00..00..00..01
```

Assigning the DS8000 LUNs to the AIX host

Use the `dscl` with the following commands to assign the new volumes (4004 - 4006 to V3), and initiate the host connections in the DS8000 (Example 8-73). The volumes and volgrp were previously created.

**Example 8-73**  Using `dscl` commands to assign storage to the AIX host

```
mkfbvol -dev IBM.2107-75L4741 -extpool P4 -name 8300_#h -type ds -cap 5 4004-4005
mkfbvol -dev IBM.2107-75L4741 -extpool P4 -name 8300_#h -type ds -cap 1 4006
mkvolgrp -dev IBM.2107-75L4741 IBM.2107-75B1981 scsimask -volume 4004-4006 V3
mkhostconnect -dev IBM.2107-75L4741 -wwname 10000000C93E007C -hosttype pSeries
    -volgrp V5 p615-tic-6_A0
mkhostconnect -dev IBM.2107-75L4741 -wwname 10000000C93E0059 -hosttype pSeries
    -volgrp V5 p615-tic-6_A1
```

Issue the `showvolgrp` command as shown in Example 8-74 to verify that LUNs are assigned to V3.

**Example 8-74**  Displaying LUNs in a volgrp using the `dscl`

```
dscl> showvolgrp -dev IBM.2107-75L4741 V3
Date/Time: March 28, 2007 12:56:45 PM CST IBM DSCLI Version: 5.2.400.304 DS:
IBM.2107-75L4741
Name p615_tic6_vg
ID V3
Type SCSI Mask
Vols 4004 4005 4006
```

Notice that LUNs 4004-4005 are assigned to volgrp V3.
Assigning and discover the DS8000 LUNs
Assign the DS8000 LUNs and run `cfgmgr` on the AIX host system.

Determining the differences between the ESS and DS8000 LUN
You can view the differences between the ESS and DS8000 by running the `lsvpcfg` command. Identify the DS8000 and ESS LUNs by the LUN ID displayed in the output of the `lsvpcfg` command as shown in Example 8-75.

Example 8-75 Identifying ESS and DS8000 LUNs

```
# lsvpcfg
vpath2 (Avail pv chuckvg) 70022665 = hdisk19 (Avail ) hdisk22 (Avail ) hdisk25 (Avail )
  hdisk34 (Avail )
vpath3 (Avail pv chuckvg) 70122665 = hdisk20 (Avail ) hdisk23 (Avail ) hdisk26 (Avail )
  hdisk35 (Avail )
vpath4 (Avail pv chuckvg) 70422665 = hdisk21 (Avail ) hdisk24 (Avail ) hdisk27 (Avail )
  hdisk36 (Avail )
vpath5 (Avail ) 75L47414005 = hdisk14 (Avail ) hdisk17 (Avail ) hdisk29 (Avail )
  hdisk32 (Avail )
vpath6 (Avail ) 75L47414006 = hdisk15 (Avail ) hdisk18 (Avail ) hdisk30 (Avail )
  hdisk33 (Avail )
vpath7 (Avail pv ) 75L47414004 = hdisk16 (Avail ) hdisk28 (Avail ) hdisk37 (Avail )
  hdisk38 (Avail )
```

The first three digits of the ESS LUN IDs are the LUN hex ID, and the last five digits are the ESS serial number. The DS8000 LUNs are the opposite: the first eight digits are the DS8000 Serial number, and the last four digits are the LUN hex ID. For example, because vpath 5 is a DS8000 LUN, 75L4741 is the serial number of the unit and 4005 is the LUN ID. The 40 is the LSS and 05 is the LUN ID in hex.

Identifying the sizes of the DS8000 LUNs
You need to know the sizes of the DS8000 LUNs to move them appropriately into the volume groups (VGs). If you use the `mklvcopy` command, make sure that there is enough space in the VG on the DS8000 LUNs collectively to create the mirrors. Issue the `bootinfo -s vpath/hdisk` command to determine the sizes in MB as shown in Example 8-76.

Example 8-76 Output showing the results of the bootinfo command

```
# bootinfo -s vpath5
5120
# bootinfo -s vpath6
1024
# bootinfo -s vpath7
5120
```

Moving the DS8000 LUNs into the VGs
Issue the `extendvg` command to move the DS8000 LUNs into the targeted volume groups as shown in Example 8-77.

Example 8-77 Using the extendvg command to move LUNs into a VG

```
# extendvg chuckvg vpath5 vpath6 vpath7
0516-1254 extendvg: Changing the PVID in the ODM.
```
Verifying that the vpaths are correctly added to the VG

Verify that the vpaths are correct with the `lsvg -p` command as shown in Example 8-78.

**Example 8-78   Output results from using the lsvg -p command**

```
# lsvg -p chuckvg
chuckvg:
  PV_NAME      PV STATE   TOTAL PPs  FREE PPs  FREE DISTRIBUTION
vpath2       active     74        0      00..00..00..00..00
vpath3       active     74        1      00..00..00..00..01
vpath4       active     14        1      00..00..00..00..01
vpath5       active     79        79     16..16..15..16..16
vpath6       active     15        15     03..03..03..03..03
vpath7       active     79        79     16..16..15..16..16
```

Determining how the LVs are spread across the vpaths

It is important to know whether you have special host LUN spreading requirements. Issue the `lslv -l` command to obtain this information as shown in Example 8-79.

**Example 8-79   Using the lslv -l command to check for host LUN spreading requirements**

```
# lslv -l ess5GBlv1
ess5GBlv1:/ess5GB1
  PV                COPIES        IN BAND       DISTRIBUTION
vpath0            037:000:000   100%          015:000:000:007:015
vpath1            036:000:000   100%          007:000:000:015:014
# lslv -l ess5GBlv2
ess5GBlv2:/ess5GB2
  PV                COPIES        IN BAND       DISTRIBUTION
vpath1            038:000:000   100%          008:015:014:000:001
vpath0            030:000:000   100%          000:008:014:008:000
vpath2            005:000:000   100%          001:000:000:003:001
# lslv -l ess1GBlv
ess1GBlv:/ess1GB
  PV                COPIES        IN BAND       DISTRIBUTION
vpath2            009:000:000   100%          002:003:002:000:002
vpath0            004:000:000   100%          000:004:000:000:000
# lslv -l loglv00
loglv00:N/A
  PV                COPIES        IN BAND       DISTRIBUTION
vpath0            001:000:000   100%          000:001:000:000:000
```

Reserving free space for an even spread of the data

Reserve free space on each LUN for an even spread of the data across LUNs. To ensure an even spread of the LV across LUNs in the VG, reserve free space on each LUN. Reserve the space by making a dummy LV on each of the targeted DS8000 LUNs using the `mklv` command (Example 8-80).

**Example 8-80   Making a dummy LV**

```
# mklv -y lvdummy chuckvg 39 vpath5
lvdummy
# extendlv lvdummy 39 vpath7
```
Verify that the dummy LV has reserved the space on the targeted LUNs by issuing the `lsvg -p` command as shown in Example 8-81.

**Example 8-81  Output from the lsvg -p command**

```
# lsvg -p chuckvg
```

<table>
<thead>
<tr>
<th>PV_NAME</th>
<th>PV STATE</th>
<th>TOTAL PPs</th>
<th>FREE PPs</th>
<th>FREE DISTRIBUTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>vpath0</td>
<td>active</td>
<td>74</td>
<td>2</td>
<td>00..02..00..00..00</td>
</tr>
<tr>
<td>vpath1</td>
<td>active</td>
<td>74</td>
<td>0</td>
<td>00..00..00..00..00</td>
</tr>
<tr>
<td>vpath2</td>
<td>active</td>
<td>14</td>
<td>0</td>
<td>00..00..00..00..00</td>
</tr>
<tr>
<td>vpath5</td>
<td>active</td>
<td>79</td>
<td>37</td>
<td>16..00..00..05..16</td>
</tr>
<tr>
<td>vpath6</td>
<td>active</td>
<td>15</td>
<td>15</td>
<td>03..03..03..03..03</td>
</tr>
<tr>
<td>vpath7</td>
<td>active</td>
<td>79</td>
<td>37</td>
<td>05..00..00..16..16</td>
</tr>
</tbody>
</table>

**Copying LV data from source LUNs to target LUNs**

Copy LV data from the ESS source LUNs to the DS8000 target LUNs using the `mklvcopy` command shown in Example 8-82.

**Example 8-82  Using the mklvcopy command**

```
# mklvcopy ess5GBlv1 2 vpath5 vpath7
```

**Verifying that the copies are correct**

Verify that the LV copies are correctly made by issuing the `lslv -l` command as shown in Example 8-83.

**Example 8-83  Verifying the copies with the lslv -l command**

```
# lslv -l ess5GBlv1
```

<table>
<thead>
<tr>
<th>PV</th>
<th>COPIES</th>
<th>IN BAND</th>
<th>DISTRIBUTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>vpath0</td>
<td>037:000:000</td>
<td>100%</td>
<td>015:000:000:007:015</td>
</tr>
<tr>
<td>vpath5</td>
<td>037:000:000</td>
<td>100%</td>
<td>016:000:000:005:016</td>
</tr>
<tr>
<td>vpath1</td>
<td>036:000:000</td>
<td>100%</td>
<td>007:000:000:015:014</td>
</tr>
<tr>
<td>vpath7</td>
<td>036:000:000</td>
<td>100%</td>
<td>005:000:000:016:015</td>
</tr>
</tbody>
</table>

Remove `lvdummy` and continue making the other LV copies by using the `rmlv` command as shown in Example 8-84.

**Example 8-84  Using the rmlv command to remove lvdummy**

```
# rmlv lvdummy
```

Warning, all data contained on logical volume lvdummy will be destroyed.

```
rm LV: Do you wish to continue? y(es) n(o)? y
```

```
rmlv: Logical volume lvdummy is removed.
```

**Synchronizing the copies**

Synchronize the copies by issuing the `syncvg -v` command as shown in Example 8-85.

**Example 8-85  Using the sync -v command**

```
# syncvg -v chuckvg
```
Checking for stale partitions
Verify that there are no stale partitions by issuing the `lsvg -l` command as shown in Example 8-86.

Example 8-86 Using the `lsvg -l` command to verify the PP state

<table>
<thead>
<tr>
<th>LV NAME</th>
<th>TYPE</th>
<th>LPs</th>
<th>PPs</th>
<th>PVs</th>
<th>LV STATE</th>
<th>MOUNT POINT</th>
</tr>
</thead>
<tbody>
<tr>
<td>ess5GBlv1</td>
<td>jfs2</td>
<td>73</td>
<td>146</td>
<td>4</td>
<td>open/syncd</td>
<td>/ess5GB1</td>
</tr>
<tr>
<td>ess5GBlv2</td>
<td>jfs2</td>
<td>73</td>
<td>146</td>
<td>5</td>
<td>open/syncd</td>
<td>/ess5GB2</td>
</tr>
<tr>
<td>ess1GBlv</td>
<td>jfs2</td>
<td>13</td>
<td>26</td>
<td>3</td>
<td>open/syncd</td>
<td>/ess1GB</td>
</tr>
<tr>
<td>loglv00</td>
<td>jfs2log</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>open/syncd</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Verifying LV copies
Verify that LV copies are shown from both the source and target LUNs for each LV by issuing the `lslv -l` command (Example 8-87).

Example 8-87 Using the `lslv -l` command to verify lv copies on the source and target LUNs

```
# lslv -l ess5GBlv1
ess5GBlv1:/ess5GB1
PV  COPIES  IN BAND      DISTRIBUTION
vpath0  037:000:000  100%  015:000:000:007:015
vpath5  037:000:000  100%  016:000:000:005:016
vpath1  036:000:000  100%  007:000:000:015:014
vpath7  036:000:000  100%  005:000:000:016:015
# lslv -l ess5GBlv2
ess5GBlv2:/ess5GB2
PV  COPIES  IN BAND      DISTRIBUTION
vpath1  038:000:000  100%  008:015:014:000:001
vpath7  036:000:000  100%  011:010:015:000:001
vpath5  036:000:000  100%  000:010:015:011:000
vpath0  030:000:000  100%  000:008:014:008:000
vpath2  005:000:000  100%  001:000:000:003:001
# lslv -l ess1GBlv
ess1GBlv:/ess1GB
PV  COPIES  IN BAND      DISTRIBUTION
vpath2  009:000:000  100%  002:003:002:000:002
vpath6  013:000:000  100%  003:003:003:003:001
vpath0  004:000:000  100%  000:004:000:000:000
# lslv -l loglv00
loglv00:N/A
PV  COPIES  IN BAND      DISTRIBUTION
vpath0  001:000:000  100%  000:001:000:000:000
vpath5  001:000:000  100%  000:001:000:000:000
```

Removing the source copies from the ESS LUNs
To remove the source copies from the ESS LUNs, issue the `rmlvcopy` command as shown in Example 8-88.

Example 8-88 Using the `rmlvcopy` command to remove the source ESS source LUNs

```
# rmlvcopy ess5GBlv1 1 vpath0 vpath1
# rmlvcopy ess5GBlv2 1 vpath0 vpath1 vpath2
```
Verifying that all the source ESS LUNs are free with no data

Verify that all the source ESS LUNs are free with no data by checking the LUNs in the VG. Issue the `lsvg -p` command as shown in Example 8-89.

**Example 8-89   Output results from the lsvg -p command**

```
# lsvg -p chuckvg
chuckvg:
   PV_NAME       PV STATE       TOTAL PPs    FREE PPs    FREE DISTRIBUTION
   vpath0        active         74           74           15..15..14..15..15
   vpath1        active         74           74           15..15..14..15..15
   vpath2        active         14           14           03..03..02..03..03
   vpath3        active         79           5            00..05..00..00..00
   vpath4        active         15           2            00..00..00..00..02
   vpath7        active         79           6            00..06..00..00..00
```

Removing the ESS source LUNs from the VGs

Remove vpath2 from the VG by issuing the `reducevg` command. Then verify that the source ESS LUN is removed from the VG by using the `lsvg -p` command as shown in Example 8-90.

**Example 8-90   An example of removing and verifying the LUN from the VG**

```
# reducevg chuckvg vpath2
# lsvg -p chuckvg
chuckvg:
   PV_NAME       PV STATE       TOTAL PPs    FREE PPs    FREE DISTRIBUTION
   vpath3        active         74           1            00..00..00..00..01
   vpath4        active         14           1            00..00..00..00..01
   vpath5        active         79           5            00..00..00..00..05
   vpath6        active         15           15           03..03..03..03..03
   vpath7        active         79           79           16..16..15..16..16
```

You can check the vpaths and associated hdisks by running the `lsvpcfg` command as shown in Example 8-91.

**Example 8-91   Results of running the lsvpcfg command**

```
# lsvpcfg
vpath2 (Avail pv) 70022665 = hdisk25 (Avail) hdisk2 (Avail) hdisk3 (Avail)
   hdisk4 (Avail)
vpath3 (Avail pv) 70122665 = hdisk20 (Avail) hdisk23 (Avail) hdisk26 (Avail)
   hdisk35 (Avail)
vpath4 (Avail pv) 70422665 = hdisk21 (Avail) hdisk24 (Avail) hdisk27 (Avail)
   hdisk36 (Avail)
vpath5 (Avail pv chuckvg) 75L47414005 = hdisk14 (Avail) hdisk17 (Avail)
   hdisk29 (Avail) hdisk32 (Avail)
vpath6 (Avail pv chuckvg) 75L47414006 = hdisk15 (Avail) hdisk18 (Avail)
   hdisk30 (Avail) hdisk33 (Avail)
```

# rmlvcopy ess1GBlv 1 vpath0 vpath2
# rmlvcopy loglv00 1 vpath0
Deleting the device definitions from the host ODM
Remove the device definitions for all the source ESS vpaths and all their associated hdisk
paths. For example, to remove vpath2 and its associated hdisk paths, issue the `rmdev`
commands as shown in Example 8-92.

Example 8-92  Deleting the device definitions

```
rmdev -dl vpath2
rmdev -dl hdisk19
rmdev -dl hdisk22
rmdev -dl hdisk25
```

In Example 8-92, only vpath2 and its associated disks were deleted. Use the same process to
delete all the vpaths and the associated hdisks owned by vpath3 and vpath4.

Verifying that the device definitions are removed
Verify that there are no defined disks left by issuing the `lsdev -Cc disk` command
(Example 8-93).

Example 8-93  Results from issuing the lsdev -Cc disk command

```
# lsdev -Cc disk
hdisk0  Available  1S-08-00-4,0  16 Bit LVD SCSI Disk Drive
hdisk1  Available  1S-08-00-5,0  16 Bit LVD SCSI Disk Drive
hdisk14 Available  1V-08-01     IBM FC 2107
hdisk15 Available  1V-08-01     IBM FC 2107
hdisk16 Available  1H-08-01     IBM FC 2107
hdisk17 Available  1H-08-01     IBM FC 2107
hdisk18 Available  1H-08-01     IBM FC 2107
hdisk28 Available  1H-08-01     IBM FC 2107
hdisk29 Available  1H-08-01     IBM FC 2107
hdisk30 Available  1H-08-01     IBM FC 2107
hdisk32 Available  1V-08-01     IBM FC 2107
hdisk33 Available  1V-08-01     IBM FC 2107
hdisk37 Available  1V-08-01     IBM FC 2107
hdisk38 Available  1V-08-01     IBM FC 2107
vpath5 Available   Data Path Optimizer Pseudo Device Driver
vpath6 Available   Data Path Optimizer Pseudo Device Driver
vpath7 Available   Data Path Optimizer Pseudo Device Driver
```

Removing the source zone definition
Remove the source zone definition from the switch fabric.
Unassigning the LUNs from the host server in the ESS

Unassign the source ESS volumes from the storage unit and verify that the ESS volumes are no longer showing on the system (Figure 8-26).

8.8 Data migration in a PowerHA clustered environment using AIX LVM mirroring

This section addresses data migration using AIX Logical Volume Manager (LVM) to migrate data from one set of IBM DS4000 LUNs to another in a PowerHA clustered environment. In this example, a file system created on a logical volume striped across four physical LUNs is migrated to a set of two physical LUNs using AIX LVM mirroring. The method for mirroring logical volumes (LV) in a clustered environment is similar to a stand-alone AIX environment. However, there are significant differences. The main difference is the way a volume group is used. In a stand-alone configuration, the volume group is a dedicated resource, whereas in a clustered configuration it is shared.

A key element of any PowerHA cluster is the data used by the highly available applications. This data is stored on AIX LVM entities. PowerHA clusters use the capabilities of the LVM to make this data accessible to multiple nodes. When you change the definition of a shared LVM component in a cluster, the operation updates the following items:

- The LVM data that describes the component on the local node
- The Volume Group Descriptor Area (VGDA) on the disks in the volume group

AIX LVM enhancements allow all nodes in the cluster to be aware of changes to a volume group, logical volume, and file system when those changes are made. Depending on the version of AIX and PowerHA software, LVM changes might not propagate automatically to all cluster nodes. Verify that the LVM definition of a volume group is the same on all cluster nodes before a failover occurs.
To change the LVM configuration on a stand-alone AIX server, you normally use AIX commands such as `extendvg` and `mklvcopy`. A shared resource in a PowerHA cluster can be reliably controlled by only one resource manager at a time. For resources defined as part of PowerHA resource groups, PowerHA must be the only resource manager controlling the resource.

Refrain from using AIX commands to modify such resources, and use only PowerHA operations on the resources. Use only Cluster Single Point Of Control (C-SPOC) operations on shared volume groups, physical disks, and file systems when the cluster is active. Using AIX commands on shared volume groups while the cluster is active can result in the volume group becoming inaccessible, and can corrupt data.

### 8.8.1 PowerHA test environment

The example PowerHA environment contains a dual-node PowerHA cluster with a DS4000 used as a shared disk subsystem. Four LUNs are presented to both cluster nodes. A volume group is created using four hdisks. The logical volume is striped across all available hdisks, and the JFS2 file system is created on the logical volume as shown in Figure 8-27.

![Figure 8-27 Data migration test PowerHA environment](image-url)
You can get information about the DS4000 LUNs using the `mpio_get_config -Av` command as shown in Example 8-94.

**Example 8-94   IBM DS4000 LUNs**

```bash
# mpio_get_config -Av
Frame id 0:
  Storage Subsystem worldwide name: 60ab80026806c000049d5867c
  Controller count: 2
  Partition count: 1
  Partition 0:
    Storage Subsystem Name = 'DRS_FASfT1'
    hdisk      LUN #   Ownership          User Label
    hdisk2         0   A (preferred)      ITSO_vol_s001
    hdisk3         8   A (preferred)      ITSO_vol_s002
    hdisk4         9   B (preferred)      ITSO_vol_s003
    hdisk5        10   A (preferred)      ITSO_vol_s004
```

Similarly, you can get information about the LVM entity definitions using the `lsvg itsovg` command (Example 8-95).

**Example 8-95   LVM entity definitions**

```bash
# lsvg itsovg
VOLUME GROUP:       itsovg VG IDENTIFIER: 0004afd800000d700000000131205383d4
VG STATE:           active                   PP SIZE:        32 megabyte(s)
VG PERMISSION:      read/write               TOTAL PPs:      2044 (65408 megabytes)
MAX LVs:            256                      FREE PPs:       2032 (65024 megabytes)
LVs:                1                        USED PPs:       12 (384 megabytes)
OPEN LVs:           1                        QUORUM:         3 (Enabled)
TOTAL PVs:          4                        VG DESCRIPTORS: 4
STALE PVs:          0                        STALE PPs:      0
ACTIVE PVs:         4                        AUTO ON:        no
Concurrent:         Enhanced-Capable         AUTO-Concurrent: Disabled
VG Mode:            Concurrent
Node ID:            3                        Active Nodes:
MAX PPs per VG:     32512
MAX PPs per PV:     1016                     MAX PVs:        32
LTG size (Dynamic): 256 kilobyte(s)          AUTO SYNC:      no
HOT SPARE:          no                       BB POLICY:      relocatable
PV RESTRICTION:     none

# lsvg -l itsovg
itsovg:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
itsolv              jfs2       12      12      4    open/syncd    /itsofs

# lslv itsolv
LLOGICAL VOLUME:     itsolv                  VOLUME GROUP:   itsovg
LV IDENTIFIER:      0004afd800000d700000000131205383d4.1 PERMISSION: read/write
VG STATE:           active/complete        LV STATE:       opened/syncd
TYPE:               jfs2                   WRITE VERIFY:   off
MAX LPs:            512                    PP SIZE:        32 megabyte(s)
COPIES:             1                      SCHED POLICY:   striped
```

Chapter 8. Using mirroring techniques  351
Data Migration to IBM Disk Storage Systems

LPs: 12  PPs: 12
STALE PPs: 0  BB POLICY: relocatable
INTER-POLICY: maximum  RELOCATABLE: no
INTRA-POLICY: middle  UPPER BOUND: 4
MOUNT POINT: /itsofs  LABEL: /itsofs
MIRROR WRITE CONSISTENCY: on/ACTIVE
EACH LP COPY ON A SEPARATE PV?: yes (superstrict)
Serialize IO?: NO
STRIPE WIDTH: 4
STRIPE SIZE: 64K

# lslv -m itsolv
itsolv:/itsofs
LP  PP  PV
0001 0104 hdisk2
0002 0104 hdisk3
0003 0104 hdisk4
0004 0104 hdisk5
0005 0105 hdisk2
0006 0105 hdisk3
0007 0105 hdisk4
0008 0105 hdisk5
0009 0106 hdisk2
0010 0106 hdisk3
0011 0106 hdisk4
0012 0106 hdisk5
0032 0111 hdisk5

# lsfs /itsofs
Name  Nodename  Mount Pt  VFS  Size  Options  Auto
Accounting  /dev/itsolv  --  /itsofs jfs2 786432 rw no no

Use the /usr/es/sbin/cluster/utilities/cltopinfo -n command to get information about PowerHA topology and resource group definitions as shown in Example 8-96.

Example 8-96  PowerHA topology and resource group definitions

# /usr/es/sbin/cluster/utilities/cltopinfo -n

NODE itso_engine1:
    Network net_diskhb_01
        itso_engine1_hdisk2_01 /dev/hdisk2
    Network net_diskhb_02
        itso_engine1_hdisk3_01 /dev/hdisk3
    Network net_ether_01
        itso_cluster_svc 9.53.11.226
        itso_engine1_boot 9.53.11.227

NODE itso_engine2:
    Network net_diskhb_01
        itso_engine2_hdisk2_01 /dev/hdisk2
    Network net_diskhb_02
        itso_engine2_hdisk3_01 /dev/hdisk3
    Network net_ether_01
        itso_cluster_svc 9.53.11.226
To migrate data from the current set of LUNs, you need to migrate a single logical volume (LV), /itsofs, containing /itso_cluster_svc file system. In the example, the second set of LUNs is allocated on the same IBM DS5000 storage subsystem. You do not need to replace or upgrade the multipathing device driver.

When both sets of LUNs are discovered on both cluster nodes, use C-SPOC commands to mirror the volume group between both sets of LUNs. After the volume group is mirrored, you can remove the mirrored copy from the original set of LUNs and remove those LUNs from the volume group.

In the example, you also perform consolidation by reducing the number of LUNs. The data from the original set of four LUNs are migrated to a set of two LUNs. The capacity of the target LUNs (32 GB) is twice the capacity of the source LUNs (16 GB).

In the example, the logical volume is striped across all available volumes within the volume group. Logical volume striping is used to improve performance by balancing the workload between physical devices. As you consolidate the data on two volumes, the width of striping is reduced from four volumes to two volumes. Verify the version of AIX you are working with before using this function because LVM striping width reduction is OS version dependent.
8.8.2 High-level migration plan using PowerHA C-SPOC LVM mirroring

Table 8-11 shows the high-level steps for a data migration using C-SPOC LVM mirroring.

<table>
<thead>
<tr>
<th>Process</th>
<th>Commands</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1. Identify the DS5000 source LUNs and LUN sizes.</td>
<td>mpio_get_config -Av bootinfo -s hdiskX</td>
<td>Gather information about the AIX host including the number of DS5000 LUNs and sizes.</td>
</tr>
<tr>
<td>Step 2. Mapping of new set of DS5000 LUNs to the cluster nodes.</td>
<td>DS5000 Storage Manager GUI</td>
<td>Use DS5000 Storage Manager to map new set of LUNs to the cluster nodes.</td>
</tr>
<tr>
<td>Step 3. Discover new DS5000 LUNs.</td>
<td>cfgmgr</td>
<td>On the cluster nodes, discover the newly assigned LUNs.</td>
</tr>
<tr>
<td>Step 4. Determine LUN sizes for both sets of LUNs</td>
<td>mpio_get_config -Av bootinfo -s hdisk#</td>
<td>Where # is the number of the hdisk.</td>
</tr>
<tr>
<td>Step 5. Include the new LUNs into the existing volume group</td>
<td>smitty cl_extendvg</td>
<td>Use the C-SPOC smitty panel to assign a PVID to a new volume and extend the volume group using the new volume. Must be repeated for all new volumes.</td>
</tr>
<tr>
<td>Step 6. Verify that the DS5000 LUNs are added to the VG on all cluster nodes.</td>
<td>lsvg -p vg_name</td>
<td>Where vg_name is the name of the VG.</td>
</tr>
<tr>
<td>Step 7. Identify the logical volumes (LVs) to migrate.</td>
<td>lsvg -l vg_name</td>
<td>Where vg_name is the name of the VG.</td>
</tr>
<tr>
<td>Step 8. Create LV copies on the new set of volumes.</td>
<td>smitty cl_mirrorvg</td>
<td>Use the C-SPOC smitty panel to create an LV mirror copy.</td>
</tr>
<tr>
<td>Step 9. Synchronize LVM mirrors.</td>
<td>smitty cl_syncvg_lv</td>
<td>Use the C-SPOC smitty panel to synchronize LVM mirrors by LV.</td>
</tr>
<tr>
<td>Step 10. Monitor the synchronization process.</td>
<td>lsvg vg_name</td>
<td>Where vg_name is the name of the VG.</td>
</tr>
<tr>
<td>Step 11. Verify LVM mirror consistency.</td>
<td>lsvg vg_name lslv lv_name mirscan -l itsolv</td>
<td>Where vg_name and lv_name are the names of VG and mirrored LV.</td>
</tr>
<tr>
<td>Step 12. Remove LV copies from the old set of volumes.</td>
<td>smitty cl_unmirrorvg</td>
<td>Use the C-SPOC smitty panel to unmirror VG.</td>
</tr>
<tr>
<td>Step 13. Remove old volumes from the VG.</td>
<td>smitty cl_reducevg lsvg -p vg_name</td>
<td>Use the C-SPOC smitty panel for cluster-wide removal of old volume from VG. The lsvg command must be run on each cluster node for verification.</td>
</tr>
<tr>
<td>Step 14. Replace communication devices in PowerHA heartbeat networks.</td>
<td>smitty cm_change_show_communication_interfaces_devices.select</td>
<td>Use the C-SPOC smitty panel to replace original heartbeat device with the devices residing on the new set of LUNs.</td>
</tr>
</tbody>
</table>
8.8.3 Detailed migration steps using PowerHA C-SPOC LVM mirroring

Migrating data from one set of LUNs to another in PowerHA clustered environments has the following detailed steps. In this example, no new device drives or multipathing software are needed because the new LUN allocation is done from the same disk subsystem.

- Identifying the DS5000 source LUNs and LUN sizes
- Mapping the new set of DS5000 LUNs to the cluster nodes
- Discovering new DS5000 LUNs
- Determining LUN sizes for both sets of LUNs
- Including the new LUNs into the existing volume group
- Verifying that the DS5000 LUNs are added to the VG
- Identifying LVs to migrate
- Creating LV copies on the new set of volumes
- Synchronizing LVM mirrors
- Monitoring the synchronization process
- Verifying LVM mirror consistency
- Removing LV copies from the old set of volumes
- Removing old volumes from the VG
- Replacing communication devices in PowerHA heartbeat networks
- Deleting the device definitions from the server ODM
- Verifying that the device definitions are removed
- Unassigning old set of DS5000 LUNs from the cluster nodes

### Identifying the DS5000 source LUNs and LUN sizes

The AIX command used to identify DS5000 LUNs depends on the firmware level of the DS5000. In the example, the firmware level of the DS5000 requires the use of MPIO multipathing on the server side. The `mpio_get_config` command shows information about the MPIO-based DS5000 subsystem and the hdisks associated with it (Example 8-97).

Specifically, the command shows the following information about the subsystem:

- The assigned name of the subsystem
- The worldwide name of the subsystem
- A list of hdisks in the Available state that are associated with the subsystem

**Example 8-97  DS5000 LUN identification**

```bash
# mpio_get_config -Av
Frame id 0:
  Storage Subsystem worldwide name: 60ab80026806c000049d5867c
  Controller count: 2
  Partition count: 1
  Partition 0:
  Storage Subsystem Name = 'DRS_FASgT1'
```

<table>
<thead>
<tr>
<th>Process</th>
<th>Commands</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 15. Delete the device definitions from the server ODM.</td>
<td><code>rmdev -dl hdisk#</code></td>
<td>Where <code>#</code> is the number of the vpath and the appropriate hdisk.</td>
</tr>
<tr>
<td>Step 16. Verify that the device definitions are removed.</td>
<td><code>mpio_get_config -Av</code> <code>lsdev -Cc disk</code></td>
<td>Check that there are no defined hdisks.</td>
</tr>
<tr>
<td>Step 17. Unassign old DS5000 volumes from all cluster nodes.</td>
<td>DS5000 Storage Manager GUI</td>
<td>Use DS5000 Storage Manager to remove mapping of old set of LUNs to the cluster nodes.</td>
</tr>
</tbody>
</table>
To determine the size of hdisks associated with DS5000 LUNs, use LVM commands that report the size of hdisks such as `lspv` and `lsvg`. These commands work when a hdisk is already included into a VG (lsvg) or has a Physical Volume Identifier (PV ID) assigned. If no PV ID is assigned to the hdisk, use the `bootinfo` command to determine the size. The `bootinfo` command reports the size of a hdisk in Megabytes (MB) as shown in Example 8-98 on page 356.

**Example 8-98  DS5000 LUN identification**

```bash
# for i in 2 3 4 5 ; do a=`bootinfo -s hdisk$i` ; echo hdisk$i capacity $a MB; done
hdisk2 capacity 16384 MB
hdisk3 capacity 16384 MB
hdisk4 capacity 16384 MB
hdisk5 capacity 16384 MB
```

**Mapping the new set of DS5000 LUNs to the cluster nodes**

In this example, the Default group host group is used as the cluster nodes, which are the only servers zoned to DS5000. The DS5000 Storage Manager GUI window is used for LUN mapping (Figure 8-28).
Discovering new DS5000 LUNs
After all new DS5000 LUNs are mapped, run `cfgmgr` on all cluster nodes to discover the newly assigned LUNs.

Determining LUN sizes for both sets of LUNs
When LUNs are created on a DS5000 disk subsystem, each LUN gets assigned a name. Use a naming convention that helps you to identify sets of LUNs. In this example, the `s` character after the second underscore in `ITSO_vol_s00#` indicates the source set. Similarly, the `t` character in `ITSO_vol_t00#` indicates the target set (Example 8-99).

Example 8-99  Identifying DS5000 LUNs

```
# /usr/bin/mpio_get_config -Av
Frame id 0:
  Storage Subsystem worldwide name: 60ab80026806c000049d5867c
  Controller count: 2
  Partition count: 1
  Partition 0:
    Storage Subsystem Name = 'DRS_FAStT1'
    hdisk      LUN #   Ownership          User Label
    hdisk2         0   A (preferred)      ITSO_vol_s001
    hdisk3         8   A (preferred)      ITSO_vol_s002
    hdisk4         9   B (preferred)      ITSO_vol_s003
    hdisk5        10   A (preferred)      ITSO_vol_s004
    hdisk6        11   B (preferred)      ITSO_vol_t001
    hdisk7        12   A (preferred)      ITSO_vol_t002
```

The newly discovered hdisk6 and hdisk7 volumes do not have PVIDs assigned yet. To verify the LUN sizes use the `bootinfo` command (Example 8-100).

Example 8-100  DS5000 LUN identification

```
# for i in 2 3 4 5 6 7;do a=`bootinfo -s hdisk$i` ;echo hdisk$i capacity $a MB; done
hdisk2 capacity 16384 MB
hdisk3 capacity 16384 MB
hdisk4 capacity 16384 MB
hdisk5 capacity 16384 MB
hdisk6 capacity 32768 MB
hdisk7 capacity 32768 MB
```

The capacity of the newly discovered hdisk6 and hdisk7 volumes is 32 GB, twice the capacity of the source volumes. The combined capacity of the new set of volumes matches the capacity of the old set, 64 GB.
Including the new LUNs into the existing volume group

To extend the existing volume group using the newly discovered volumes, use C-SPOC commands to maintain consistency across all cluster nodes. The easiest way to start C-SPOC commands is to use the AIX smitty interface (Example 8-101). Unlike standard AIX LVM commands, C-SPOC commands require a number of additional parameters such as resource group name and node names. Although it is possible to add all those parameters in a command line, it is easier to do through smitty.

Example 8-101  Extending itsovg volume group using C-SPOC smitty interface “smitty cl Extendvg”

+--------------------------------------------------------------------------+
| Select the Volume Group that will hold the new Logical Volume            |
|                                                                          |
| Move cursor to desired item and press Enter. Use arrow keys to scroll.  |
|                                                                          |
| #Volume Group  Resource Group  Node List                                 |
| itsovg          itso_rg        itso_engine1,itso_engine2                  |
| F1=Help         F2=Refresh     F3=Cancel                                  |
| F8=Image        F10=Exit       Enter=Do                                   |
| /=Find          n=Find Next                                            |
+--------------------------------------------------------------------------+

+--------------------------------------------------------------------------+
| Physical Volume Names                                                    |
|                                                                          |
| Move cursor to desired item and press Enter.                             |
|                                                                          |
| 0004afca24cd59c3 ( hdisk6 on all cluster nodes )                         |
| 0004afca24cd5ae2 ( hdisk7 on all cluster nodes )                         |
| F1=Help         F2=Refresh     F3=Cancel                                  |
| F8=Image        F10=Exit       Enter=Do                                   |
| /=Find          n=Find Next                                            |
+--------------------------------------------------------------------------+

Add a Volume to a Volume Group

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[Entry Fields]

<table>
<thead>
<tr>
<th>VOLUME GROUP name</th>
<th>itsovg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource Group Name</td>
<td>itso_rg</td>
</tr>
<tr>
<td>Node List</td>
<td>itso_engine1,itso_engine2</td>
</tr>
<tr>
<td>Reference node</td>
<td>itso_engine1</td>
</tr>
<tr>
<td>VOLUME names</td>
<td>hdisk6</td>
</tr>
<tr>
<td>F1=Help</td>
<td>F2=Refresh</td>
</tr>
<tr>
<td>F5=Reset</td>
<td>F6=Command</td>
</tr>
<tr>
<td>F9=Shell</td>
<td>F10=Exit</td>
</tr>
<tr>
<td>F10=Exit</td>
<td>Enter=Do</td>
</tr>
<tr>
<td>F4=List</td>
<td>F8=Image</td>
</tr>
<tr>
<td>F7=Edit</td>
<td></td>
</tr>
</tbody>
</table>
The `smitty` interface prompts you to select the volume group to expand and the volume to add to the selected volume group. It then determines the list of cluster nodes and the reference node. The reference node is the node where the `itso_rg` resource group is online. C-SPOC also performs consistency checking along the way.

If the PV IDs do not match, the interface does not allow you to expand the volume group until the inconsistency is resolved. The simplest resolution is to clear the PVIDs on hdisk6 and hdisk7 on all cluster nodes using the `chdev -l hdisk# -a pv=clear` command. You must go through this process once for every disk used for expanding the volume group.

You can also use command-line interface (CLI) to achieve the same result. However, if you use the CLI, you must manually check PV ID consistency. The advantage of using the CLI is that you can automate this task. If you are adding many volumes, CLI can save you time. A CLI command equivalent to the set of `smitty` panels shown in Example 8-101 on page 358 is shown in Example 8-102.

```
Example 8-102  C-SPOC CLI command for expanding VG

# /usr/es/sbin/cluster/sbin/cl_extendvg -cspoc -n'itso_engine1,itso_engine2'
     -R'itso_engine1' itsovg hdisk6
```

The C-SPOC command can be run from any node in the cluster, regardless of which node has the resource group online.

**Verifying that the DS5000 LUNs are added to the VG**

To verify that the hdisk6s are added to the VG correctly, run the `lsvg -p` command on each cluster node as shown in Example 8-103.

```
Example 8-103  Output results from using the lsvg -p command

<itso_engine1>
# lsvg -p itsovg
itsovg:
   PV_NAME   PV STATE TOTAL PPs   FREE PPs FREE DISTRIBUTION
hdisk2   active    511      508  103..99..102..102..102
hdisk3   active    511      508  103..99..102..102..102
hdisk4   active    511      508  103..99..102..102..102
hdisk5   active    511      508  103..99..102..102..102
hdisk7   active    1023    1023  205..205..204..204..205
hdisk6   active    1023    1023  205..205..204..204..205

<itso_engine2>
# lsvg -p itsovg
itsovg:
   PV_NAME   PV STATE TOTAL PPs   FREE PPs FREE DISTRIBUTION
hdisk2   active    511      508  103..99..102..102..102
hdisk3   active    511      508  103..99..102..102..102
hdisk4   active    511      508  103..99..102..102..102
hdisk5   active    511      508  103..99..102..102..102
hdisk7   active    1023    1023  205..205..204..204..205
hdisk6   active    1023    1023  205..205..204..204..205
```

In this example, both cluster nodes (`itso_engine1` and `itso_engine2`) have the `itsovg` VG successfully expanded with the newly discovered `hdisk6` and `hdisk7`. 

---

Chapter 8. Using mirroring techniques 359
Identifying LVs to migrate

To identify the LVs to migrate from the source set of LUNs, issue the command `lsvg -l` on all cluster nodes as shown in Example 8-104.

Example 8-104 The output results of issuing the lsvg -l command against a VG

<table>
<thead>
<tr>
<th>LV NAME</th>
<th>TYPE</th>
<th>LPs</th>
<th>PPs</th>
<th>PVs</th>
<th>LV STATE</th>
<th>MOUNT POINT</th>
</tr>
</thead>
<tbody>
<tr>
<td>itsolv</td>
<td>jfs2</td>
<td>12</td>
<td>12</td>
<td>4</td>
<td>open/syncd</td>
<td>/itsofs</td>
</tr>
</tbody>
</table>

LV `itsolv` is in open/syncd state on one of the cluster nodes and closed/syncd on the other. In PowerHA, cluster JFS2 LV can be opened on only one node at a time. After the file system is unmounted, the LV switches into closed/syncd state and can be opened by another node. Attempts to open LV on a cluster node while it is opened on another node results in an error message. AIX LVM uses locking mechanisms to prevent from more than one node mounting the same JFS2 file system at a time.

In this example, the JFS2 file system is configured with INLINE JFS2 log. There is no separate JFS2 log LV device associated with the `itsofs` file system. If there was a separate JFS2 log LV, it must be migrated to the new set of LUNs along with the JFS2 LV. By default the JFS2 file system is configured with a separate jfs2log LV. In this example INLINE Jfs2 is used to keep the configuration simple. In this configuration, you only need to migrate `itsolv` LV.

Creating LV copies on the new set of volumes

The C-SPOC tool used for mirroring the VG creates an extra copy of all LVs within a VG. You cannot mirror only some of the LVs in a VG with C-SPOC tools. The standard AIX LVM commands provide more flexibility for selective mirroring of LVs. Mirroring of all LVs in a VG is adequate for the task of migrating data from one set of LUNs to another.

To create LV copies of all LVs in the `itsovg` VG, use the C-SPOC `smitty` interface. The `smitty` interface offers selection choices and performs all the necessary checking on all cluster nodes before running LVM commands. If you are mirroring data in a few VGs, use the `smitty` interface as shown in Example 8-105.

Example 8-105 Mirroring `itsovg` VG using the C-SPOC `smitty` interface

<table>
<thead>
<tr>
<th>#Volume Group</th>
<th>Resource Group</th>
<th>Node List</th>
</tr>
</thead>
<tbody>
<tr>
<td>itsovg</td>
<td>itso_rg</td>
<td>itso_engine1,itso_engine2</td>
</tr>
</tbody>
</table>

Select the Volume Group to Mirror

Move cursor to desired item and press Enter. Use arrow keys to scroll.

F1=Help       F2=Refresh       F3=Cancel
F8=Image      F10=Exit         Enter=Do
=Find          n=Find Next
Chapter 8. Using mirroring techniques

Physical Volume Names

Move cursor to desired item and press F7.
ONE OR MORE items can be selected.
Press Enter AFTER making all selections.

Auto-select
drs_engine1  hdisk2
> drs_engine1  hdisk6
> drs_engine1  hdisk7
drs_engine1  hdisk3
drs_engine1  hdisk4
drs_engine1  hdisk5

F1=Help               F2=Refresh              F3=Cancel
F7=Select             F8=Image                F10=Exit
Enter=Do              /=Find                  n=Find Next

Mirror a Volume Group

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[Entry Fields]

* VOLUME GROUP name                                   itsovg
Resource Group Name                                 itso_rg
Node List                                           itso_engine1,itso_engine2
Reference node                                      itso_engine1
PHYSICAL VOLUME names                               hdisk6 hdisk7
Mirror sync mode                                    No Sync                  +
Number of COPIES of each logical                   2                        +
    partition
Keep Quorum Checking On?                            no                       +
Create Exact LV Mapping?                            no                       +

F1=Help             F2=Refresh           F3=Cancel           F4=List
F5=Reset            F6=Command           F7=Edit             F8=Image
F9=Shell            F10=Exit             Enter=Do

After you select the VG to mirror and the set of hdisks to create LV copy on, you are presented
with choices for various attributes. Leave all attributes on the default values except for Mirror
sync mode. Change this attribute value to No Sync. This parameter illustrates the VG
configuration change before data synchronization takes place. In addition, changing this
attribute if you want to increase the parallelism of the data synchronization process. See
“Synchronizing LVM mirrors” on page 363 for more details about the LV synchronization
process.
Verify the results of creating LV copies on the new set of LUNs using the `lsvg` command as shown in Example 8-106.

**Example 8-106   Results of creating LV copies**

```
# lsvg itsovg
VOLUME GROUP:     itsovg VG IDENTIFIER:  0004afd80000d70000000131205383d4
VG STATE:         active                  PP SIZE:   32 megabyte(s)
VG PERMISSION:    passive-only            TOTAL PPs: 4090 (130880 megabytes)
MAX LVs:          256                     FREE PPs:  4066 (130112 megabytes)
LVs:              1                        USED PPs:   24 (768 megabytes)
OPEN LVs:         0                        QUORUM:    1 (Disabled)
TOTAL PVs:        6                        VG DESCRIPTORS: 6
STALE PVS:        2                        STALE PPs:  12
ACTIVE PVs:       6                        AUTO ON:   no
Concurrent:       Enhanced-Capable          Auto-Concurrent: Disabled
VG Mode:          Concurrent                Node ID:    3
Max PVs per VG:   32512
Max PPs per PV:   2032                     MAX PVs:  16
LTG size (Dynamic): 256 kilobyte(s)       AUTO SYNC:  no
HOT SPARE:        no                       BB POLICY: relocatable
PV RESTRICTION:   none                      Inter-Policy: maximum
                  AUTO ON:   no
                  BB POLICY: relocatable
                  INTER-POLICY: maximum
                  INTRA-POLICY: middle
                  UPPER BOUND: 4
                  MOUNT POINT: /itsofs
                  LABEL:      /itsofs
```

As you can see in Example 8-106 on page 362, you now have a logical volume with two copies, two stale physical volumes (PVs), and 12 stale physical partitions (PPs). Stale volumes and stale partitions are present because you delayed the data synchronization. The AIX command that scans all LV copies and identifies stale PVs is `mirscan` (Example 8-107).

The `mirscan` command can run only on a cluster node where the `itso_rg` resource group is online.

**Example 8-107   Identifying stale PVs**

```
# mirscan -l itsolv
START TIME: Thu Jul 14 00:03:50 CEST:2011
OP STATUS PVNAME PP SYNC IOFAIL LVNAME LP CP TARGETPV
s SUCCESS hdisk2 104 synced no itsolv 1 1
```
The hdisk6 and hdisk7 physical volumes are in stale state as indicated in the SYNC column. These PVs contain the second copy of the LV as shown in the CP column. FAILURE in the STATUS column indicates that the partition was not synchronized and is still incapable of performing I/O operations because it contains no valid data.

Synchronizing LVM mirrors

Synchronize LV copies using the C-SPOC smitty interface as shown in Example 8-108. Synchronization can be done either for one LV at a time, or for all LVs in a single VG. In this example, it makes no difference because there is only one LV to synchronize. A VG-wide synchronization process, in most cases, is adequate for data migration.

Example 8-108  LV copy synchronization using C-SPOC smitty interface "smitty cl_syncvg_lv"
Synchronize LVM Mirrors by Logical Volume

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[Entry Fields]

LOGICAL VOLUME name                                 itsolv
VOLUME GROUP name                                   itsovg
Resource Group Name                                 itso_rg
* Node List                                           itso_engine1, itso_engine2

Number of Partitions to Sync in Parallel           [2]                        +#
Synchronize All Partitions                          no                        +
Delay Writes to VG from other cluster nodes during  no                        +
this Sync

As smitty takes you through the three panels, it prompts you to select the VG and LV to
synchronize, and determines the list of cluster nodes.

In the third smitty panel, select a value for the Number of Partitions to Sync in Parallel
attribute. This attribute specifies the number of Logical Partitions (LPs) that are synchronized
in parallel. The valid range is 1 - 32. The higher the number, the higher the synchronization
rate.

However, consider the performance impact to avoid setting this attribute too high. Find a
balance between acceptable performance impact and the synchronization rate. If you are
unsure, do not change the default attribute value. This value allows the synchronization
process run with minimal performance impact, but it takes the longest to complete.

In this example, the LVM mirror synchronization is started with 2 LPs synchronized in parallel.
The rate depends on the following factors:

- System type
- Availability of system resources
- Number of physical disks in the volume group
- Performance of the disk subsystem
- Load on the disk subsystem
- Many other factors

Understanding the environment is essential in determining the optimal synchronization rate.
Monitoring the synchronization process
To monitor the synchronization process, use the output of `lsvg itsvg` command. The STALE PPs field shows the number of PPs that are not synchronized yet. Knowing the size of the PP, 32 MB in the example, you can calculate the capacity that the unsynchronized PPs occupy. The same method can be used to estimate the duration of the process. Run the command twice with a few minutes interval between runs. Calculate how many PPs were synchronized between the runs. Divide the time between the two runs by the number of synchronized PPs to get the synchronization rate. To estimate the completion time, divide the STALE PPs number by the calculated rate. For example, if the calculated synchronization rate is 12 PPs per minute and the number of STALE PPs is 120, it takes approximately 10 minutes to complete the synchronization.

Verifying LVM mirror consistency
To verify that LVM mirrors are fully synchronized, run the normal AIX LVM commands and look for stale PVs and PPs in the VG (Example 8-109). If there are remaining stale PVs/PPs, investigate further.

Example 8-109   Verification of the state of VG and LV

```
# lsvg itsvg
VOLUME GROUP:       itsvg VG IDENTIFIER: 0004afd80000d700000000131205383d4
VG STATE:           active                   PP SIZE: 32 megabyte(s)
VG PERMISSION:      read/write               TOTAL PPs: 4090 (130880 megabytes)
MAX LVs:            256                      FREE PPs: 4066 (130112 megabytes)
LVs:                1                        USED PPs:       24 (768 megabytes)
OPEN LVs:           1                        QUORUM:         1 (Disabled)
TOTAL PVs:          6                        VG DESCRIPTORS: 6
STALE PVs:          0
STALE PPs:      0
ACTIVE PVs:         6                        AUTO ON:        no
Concurrent:         Enhanced-Capable         Auto-Concurrent: Disabled
VG Mode:            Concurrent
Node ID:            1                        Active Nodes:       3
MAX PPs per VG:     32512
MAX PPs per PV:     2032                     MAX PVs:        16
LTG size (Dynamic): 256 kilobyte(s)          AUTO SYNC:      no
HOT SPARE:          no                       BB POLICY:      relocatable
PV RESTRICTION:     none

# lsvg -l itsvg
itsvg:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
itsolv              jfs2       12      24      6    closed/syncd  /itsofs

# lslv itsolv
LOGICAL VOLUME:     itsolv                 VOLUME GROUP:   itsvg
LV IDENTIFIER:      0004afd80000d700000000131205383d4.1 PERMISSION:  read/write
VG STATE:           active/complete        LV STATE:       opened/syncd
TYPE:               jfs2                   WRITE VERIFY:   off
MAX LPs:            512                    PP SIZE:        32 megabyte(s)
COPIES:             2                      SCHED POLICY:   striped
LPs:                12                     PPs:            24
STALE PPs:          0
STALE PPs:          0
INTER-POLICY:       maximum                BB POLICY:      relocatable
INTRA-POLICY:       middle                 UPPER BOUND:    4
MOUNT POINT:        /itsofs                LABEL:          /itsofs
```
MIRROR WRITE CONSISTENCY: on/ACTIVE
EACH LP COPY ON A SEPARATE PV ?: yes (superstrict)
Serialize IO ?: NO
STRIPE WIDTH: 4
STRIPE SIZE: 64K

The results of the verification commands should be identical on all cluster nodes. In the example, no stale PVs and PPs are found, indicating that the LVM mirrors are fully in sync. For additional validation, run the **mirscan** command on the reference node. However, it takes significantly longer for the **mirscan** command to complete report generation because it examines each allocated partition on the specified device.

Example 8-110 shows a **mirscan** report of fully synchronized LVM mirrors for reference purposes. The STATUS of physical volumes *hdisk6* and *hdisk7* is SUCCESS, and SYNC is synced.

```
Example 8-110   Examining allocated partitions with mirscan

# mirscan -l itsolv
START TIME: Thu Jul 14 09:06:15 CEST:2011

<table>
<thead>
<tr>
<th>OP</th>
<th>STATUS</th>
<th>PVNAME</th>
<th>PP</th>
<th>SYNC</th>
<th>IOFAIL</th>
<th>LVNAME</th>
<th>LP</th>
<th>CP</th>
<th>TARGETPV</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SUCCESS</td>
<td>hdisk2</td>
<td>104 synced</td>
<td>no</td>
<td>itso1v</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SUCCESS</td>
<td>hdisk2</td>
<td>105 synced</td>
<td>no</td>
<td>itso1v</td>
<td>5</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SUCCESS</td>
<td>hdisk2</td>
<td>106 synced</td>
<td>no</td>
<td>itso1v</td>
<td>9</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SUCCESS</td>
<td>hdisk3</td>
<td>104 synced</td>
<td>no</td>
<td>itso1v</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SUCCESS</td>
<td>hdisk3</td>
<td>105 synced</td>
<td>no</td>
<td>itso1v</td>
<td>6</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SUCCESS</td>
<td>hdisk3</td>
<td>106 synced</td>
<td>no</td>
<td>itso1v</td>
<td>10</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SUCCESS</td>
<td>hdisk4</td>
<td>104 synced</td>
<td>no</td>
<td>itso1v</td>
<td>3</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SUCCESS</td>
<td>hdisk4</td>
<td>105 synced</td>
<td>no</td>
<td>itso1v</td>
<td>7</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SUCCESS</td>
<td>hdisk4</td>
<td>106 synced</td>
<td>no</td>
<td>itso1v</td>
<td>11</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SUCCESS</td>
<td>hdisk5</td>
<td>104 synced</td>
<td>no</td>
<td>itso1v</td>
<td>4</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SUCCESS</td>
<td>hdisk5</td>
<td>105 synced</td>
<td>no</td>
<td>itso1v</td>
<td>8</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SUCCESS</td>
<td>hdisk5</td>
<td>106 synced</td>
<td>no</td>
<td>itso1v</td>
<td>12</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SUCCESS</td>
<td>hdisk7</td>
<td>206 synced</td>
<td>no</td>
<td>itso1v</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SUCCESS</td>
<td>hdisk7</td>
<td>207 synced</td>
<td>no</td>
<td>itso1v</td>
<td>4</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SUCCESS</td>
<td>hdisk7</td>
<td>208 synced</td>
<td>no</td>
<td>itso1v</td>
<td>6</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SUCCESS</td>
<td>hdisk7</td>
<td>209 synced</td>
<td>no</td>
<td>itso1v</td>
<td>8</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SUCCESS</td>
<td>hdisk7</td>
<td>210 synced</td>
<td>no</td>
<td>itso1v</td>
<td>10</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SUCCESS</td>
<td>hdisk7</td>
<td>211 synced</td>
<td>no</td>
<td>itso1v</td>
<td>12</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SUCCESS</td>
<td>hdisk6</td>
<td>206 synced</td>
<td>no</td>
<td>itso1v</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SUCCESS</td>
<td>hdisk6</td>
<td>207 synced</td>
<td>no</td>
<td>itso1v</td>
<td>3</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SUCCESS</td>
<td>hdisk6</td>
<td>208 synced</td>
<td>no</td>
<td>itso1v</td>
<td>5</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SUCCESS</td>
<td>hdisk6</td>
<td>209 synced</td>
<td>no</td>
<td>itso1v</td>
<td>7</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SUCCESS</td>
<td>hdisk6</td>
<td>210 synced</td>
<td>no</td>
<td>itso1v</td>
<td>9</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SUCCESS</td>
<td>hdisk6</td>
<td>211 synced</td>
<td>no</td>
<td>itso1v</td>
<td>11</td>
<td>2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

END TIME: Thu Jul 14 09:09:13 CEST:2011
```

Removing LV copies from the old set of volumes
After you confirm that all LV copies are fully synchronized, remove the copy of the LV on the old set of volumes. This process is sometimes called unmirroring.

The C-SPOC tool used for unmirroring the VG removes an extra copy of all LVs within a VG. You cannot unmirror only some of the LVs in a VG. In this case, use the standard AIX LVM commands. For data migration, however, unmirroring of all LVs in a VG is adequate.
Before removing LV copies, consider your backout approach. When the LV is mirrored, you effectively have multiple (up to three) physical copies of your data. There are two ways you can “break” LV mirroring. One is to remove and discard the extra copy. The second is to split that extra copy and form an independent LV. The second approach is addressed in more detail in 8.8.4, “Migration Backout Considerations” on page 375.

This example uses the first approach of removing and discarding the LV copy on the old set of volumes: hdisk2 to hdisk5. Use the `lslv -m` command to view the relationship between PVs and LV copies as shown in Example 8-111.

**Example 8-111  Relationship between LV copies and PVs**

```
# lslv -m itsolv

itsolv:/itsofs

<table>
<thead>
<tr>
<th>LP</th>
<th>PP1</th>
<th>PV1</th>
<th>PP2</th>
<th>PV2</th>
<th>PP3</th>
<th>PV3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001</td>
<td>0104</td>
<td>hdisk2</td>
<td>0206</td>
<td>hdisk6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0002</td>
<td>0104</td>
<td>hdisk3</td>
<td>0206</td>
<td>hdisk7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0003</td>
<td>0104</td>
<td>hdisk4</td>
<td>0207</td>
<td>hdisk6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0004</td>
<td>0104</td>
<td>hdisk5</td>
<td>0207</td>
<td>hdisk7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0005</td>
<td>0105</td>
<td>hdisk2</td>
<td>0208</td>
<td>hdisk6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0006</td>
<td>0105</td>
<td>hdisk3</td>
<td>0208</td>
<td>hdisk7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0007</td>
<td>0105</td>
<td>hdisk4</td>
<td>0209</td>
<td>hdisk6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0008</td>
<td>0105</td>
<td>hdisk5</td>
<td>0209</td>
<td>hdisk7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0009</td>
<td>0106</td>
<td>hdisk2</td>
<td>0210</td>
<td>hdisk6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0010</td>
<td>0106</td>
<td>hdisk3</td>
<td>0210</td>
<td>hdisk7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0011</td>
<td>0106</td>
<td>hdisk4</td>
<td>0211</td>
<td>hdisk6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0012</td>
<td>0106</td>
<td>hdisk5</td>
<td>0211</td>
<td>hdisk7</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

PP1 is the first copy of the LV, PP2 is the second, and PP3 is the third copy. The first copy of the LV is on the old set of volumes, hdisk2 to hdisk5 in the PV1 column. Before these volumes can be excluded from the VG, remove the LV copy on these volumes.

To remove the LV copies of all LVs in `itsovg` VG, use the C-SPOC `smitty` interface (Example 8-112). `Smitty` offers selection choices and performs all the necessary checking on all cluster nodes before running LVM commands. If you are unmirroring data on a few VGs, use the C-SPOC `smitty` interface to simplify the process.

**Example 8-112  Unmirroring itsovg VG using C-SPOC smitty interface “smitty cl_unmirrorvg”**

```
+--------------------------------------------------------------------------+
| Select the Volume Group to Unmirror                                       |
| Move cursor to desired item and press Enter.                             |
| #Volume Group    Resource Group             Node List                   |
| itsovg          itso_rg                    itso_engine1,itso_engine2   |
| F1=Help                 F2=Refresh              F3=Cancel                |
| F8=Image                F10=Exit                Enter=Do                 |
| /=Find                  n=Find Next                                      |
+--------------------------------------------------------------------------+

+--------------------------------------------------------------------------+
| Physical Volume Names                                                   |
| Move cursor to desired item and press F7.                                |
| ONE OR MORE items can be selected.                                      |
+--------------------------------------------------------------------------+
Unmirror a Volume Group

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[Entry Fields]

VOLUME GROUP name itsovg
Resource Group Name itso_rg
Node List itso_engine1, itso_engine2
Reference node itso_engine1
PHYSICAL VOLUME names hdisk2, hdisk3, hdisk4, hdisk5

Number of COPIES of each logical partition 1

F1=Help F2=Refresh F3=Cancel F4=List
F5=Reset F6=Command F7=Edit F8=Image
F9=Shell F10=Exit Enter=Do

On the second smitty panel, select the set of PVs where the first copy of the LV is located. If you do not select the PVs, the tool removes the most recent copy of the LV, effectively deleting the result of the migration. After you confirm the choices on the last smitty panel, the LV copy is removed. To confirm the LV copy is removed, use lsv command as shown in Example 8-113.

Example 8-113 Verifying LV copy removal

# lsv -m itsolv
itsolv:/itsofs

<table>
<thead>
<tr>
<th>PP1</th>
<th>PV1</th>
<th>PP2</th>
<th>PV2</th>
<th>PP3</th>
<th>PV3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001</td>
<td>0206</td>
<td>hdisk6</td>
<td>0002</td>
<td>0206</td>
<td>hdisk7</td>
</tr>
<tr>
<td>0003</td>
<td>0207</td>
<td>hdisk6</td>
<td>0004</td>
<td>0207</td>
<td>hdisk7</td>
</tr>
<tr>
<td>0005</td>
<td>0208</td>
<td>hdisk6</td>
<td>0006</td>
<td>0208</td>
<td>hdisk7</td>
</tr>
<tr>
<td>0007</td>
<td>0209</td>
<td>hdisk6</td>
<td>0008</td>
<td>0209</td>
<td>hdisk7</td>
</tr>
<tr>
<td>0009</td>
<td>0210</td>
<td>hdisk6</td>
<td>0010</td>
<td>0210</td>
<td>hdisk7</td>
</tr>
</tbody>
</table>
The LV now has only one copy on the new set of PVs, `hdisk6` and `hdisk7`. To confirm that no PPs are still used on the old set of LUNs, verify the USED PPs attribute. These attributes can be viewed in the `lspv` command output (Example 8-114).

**Example 8-114  Used PPs validation**

```
# lspv hdisk2
PHYSICAL VOLUME:    hdisk2                   VOLUME GROUP:     itsovg
PV IDENTIFIER:      0004afca0b8dedbd VG IDENTIFIER
0004afd80000d70000000131205383d4
PV STATE:           active
STALE PARTITIONS:   0                        ALLOCATABLE:      yes
PP SIZE:            32 megabyte(s)           LOGICAL VOLUMES:  0
TOTAL PPs:          511 (16352 megabytes)    VG DESCRIPTORS:   1
FREE PPs:           511 (16352 megabytes)    FREE DISTRIBUTION: 103..102..102..102..102
USED PPs:           0 (0 megabytes)          USED DISTRIBUTION: 00..00..00..00..00
MIRROR POOL:        None

Removing old volumes from the VG

After you verify that no PPs are still used on the old set of PVs, those volumes can be excluded from the VG.

To remove the old volumes, use the C-SPOC commands to maintain consistency across all cluster nodes. The removal is done using the `smitty` interface as shown in Example 8-115.

**Example 8-115  Reducing itsovg VG using C-SPOC smitty interface “smitty cl_reducevg”**

```
+--------------------------------------------------------------------------+
|      Select the Volume Group that will hold the new Logical Volume       |
|                                                                          |
| Move cursor to desired item and press Enter. Use arrow keys to scroll.   |
|                                                                          |
|   #Volume Group    Resource Group             Node List                  |
|    itsovg          itso_rg                    itso_engine1,itso_engine2 |
|                                                                          |
| F1=Help                 F2=Refresh              F3=Cancel                |
| F8=Image                F10=Exit                Enter=Do                 |
| /=Find                  n=Find Next                                      |
+--------------------------------------------------------------------------+

+--------------------------------------------------------------------------+
|                          Physical Volume Names                             |
|                                                                          |
| Move cursor to desired item and press Enter.                            |
|                                                                          |
| drs_engine1     hdisk2                                                 |
| drs_engine1     hdisk6                                                 |
| drs_engine1     hdisk7                                                 |
| drs_engine1     hdisk3                                                 |
| drs_engine1     hdisk4                                                 |
+--------------------------------------------------------------------------+
Remove a Volume from a Volume Group

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[Entry Fields]

VOLUME GROUP name
Resource Group Name
Node List
VOLUME names
Reference node
FORCE deallocation of all partitions on this physical volume?

On the second smitty panel, select a single PV for exclusion. The same process needs to be repeated for every volume of the old set. Exclusion of the volume must be done only on one node. The update is then propagated by C-SPOC to all nodes in the cluster.

To verify that all the volumes from the old set are excluded from the VG, run the \texttt{lspv} command on the cluster nodes (Example 8-116). Configuration on all the cluster nodes should be consistent.

\textbf{Example 8-116} Verifying consistency of PVs configuration of PowerHA cluster nodes

\begin{verbatim}
# lspv
hdisk2  0004afca0b8dedbd  None
hdisk6  0004afca24cd59c3  itsovg  concurrent
hdisk7  0004afca24cd5ae2  itsovg  concurrent
hdisk0  0004afca82d256bb  old_rootvg
hdisk1  0004afca95554544  rootvg  active
hdisk3  0004afca0b8e068a  None
hdisk4  0004afca0b8e136c  None
hdisk5  0004afca0b8e1f63  None

# lspv
hdisk2  0004afca0b8dedbd  None
hdisk6  0004afca24cd59c3  itsovg  concurrent
hdisk7  0004afca24cd5ae2  itsovg  concurrent
hdisk0  0004afdb7db8511f  old_rootvg
hdisk1  0004afdb9555a5e21  rootvg  active
hdisk3  0004afca0b8e068a  None
\end{verbatim}
All PVs from the old set of LUNs are now removed from the `itsovg` VG configuration.

### Replacing communication devices in PowerHA heartbeat networks

If some of the volumes in the old LUN set were configured as communication devices (in disk heartbeat networks), those devices must be removed before a corresponding hdisk device can be removed. After the old communication devices are removed, new communication devices must be configured on the new set of LUNs. If you try to remove old hdisks before removing the corresponding communication devices, an error message indicates that the specified device is busy and will not be removed. As shown in Example 8-117, the `/dev/hdisk2` device is configured as communication interface in the `net_diskhb_01` network, and `/dev/hdisk3` in `net_diskhb_02`.

**Example 8-117  PowerHA topology information**

```
# /usr/es/sbin/cluster/utilities/cltopinfo -n

NODE itso_engine1:
    Network net_diskhb_01
      itso_engine1_hdisk2_01 /dev/hdisk2
    Network net_diskhb_02
      itso_engine1_hdisk3_01 /dev/hdisk3
    Network net_ether_01
      itso_cluster_svc 9.53.11.226
      itso_engine1_boot 9.53.11.227

NODE itso_engine2:
    Network net_diskhb_01
      itso_engine2_hdisk2_01 /dev/hdisk2
    Network net_diskhb_02
      itso_engine2_hdisk3_01 /dev/hdisk3
    Network net_ether_01
      itso_cluster_svc 9.53.11.226
      itso_engine2_boot 9.53.11.228
```

In the example, two heartbeat networks are configured. Update one heartbeat network at a time so that one heartbeat network is always available.

To remove a communication device from a heartbeat network, use the C-SPOC interface as shown in Example 8-118. The same result can be achieved using the `smitty` C-SPOC interface.

**Example 8-118  C-SPOC communication device removal using C-SPOC CLI interface**

```
# /usr/es/sbin/cluster/utilities/clrmnode -a'itso_engine2_hdisk2_01'
WARNING: Serial network [net_diskhb_01] has 1 communication device(s) configured. Two devices are required for a serial network.

# /usr/es/sbin/cluster/utilities/clrmnode -a'itso_engine1_hdisk2_01'
Network removed: net_diskhb_01
```

---

Chapter 8. Using mirroring techniques  371
After one pair of communication devices and the corresponding heartbeat network are removed, you can add another pair of communication devices, one per cluster node. This new pair must be configured using hdisk devices on the new set of LUNs.

To configure the new communication devices and heartbeat network correctly, use the C-SPOC smitty interface on all participating cluster nodes. The interface allows you to select devices and validates consistency across cluster nodes. To get to the smitty panel that defines the heartbeat communication devices, perform the following steps:

1. Start the interface using the smitty hacmp command.
3. Select a communication device candidate as shown in Example 8-119.

Example 8-119  C-SPOC smitty interface for creating disk heartbeat network

```
+--------------------------------------------------------------------------+
| Select Point-to-Point Pair of Discovered Communication Devices to Add   |
|                                                                          |
| Move cursor to desired item and press F7.                               |
| ONE OR MORE items can be selected.                                     |
| Press Enter AFTER making all selections.                               |
|                                                                          |
|   # Node                              Device   Pvid                      |
|     drs_engine1                       hdisk2   0004afca346ba911          |
|     drs_engine2                       hdisk2   0004afca346ba911          |
|     drs_engine1                       hdisk4   0004afca356ea2c6          |
|     drs_engine2                       hdisk4   0004afca356ea2c6          |
|     drs_engine1                       hdisk5   0004afca346ba720          |
|     drs_engine2                       hdisk5   0004afca346ba720          |
|     drs_engine1                       hdisk6   0004afca346ba892          |
|     drs_engine2                       hdisk6   0004afca346ba892          |
|                                                                          |
| F1=Help                 F2=Refresh              F3=Cancel                |
| F7=Select               F8=Image                F10=Exit                 |
| Enter=Do                /=Find                  n=Find Next              |
+--------------------------------------------------------------------------+
```

Repeat this process for any other heartbeat networks. After the reconfiguration of net_diskhb_02, the test cluster topology configuration looks as shown in Example 8-120.

Example 8-120  PowerHA cluster topology after replacing communication devices

```
# /usr/es/sbin/cluster/utilities/cltopinfo -n

NODE drs_engine1:
  Network net_diskhb_01
    drs_engine1_hdisk6_01 /dev/hdisk6
  Network net_diskhb_02
    drs_engine1_hdisk7_01 /dev/hdisk7
  Network net_ether_01
    drs_cluster_svc 9.153.1.226
    drs_engine1_boot 9.153.1.227
```
The communication device replacement is now complete for both heartbeat networks.

The last step is to propagate the updates from the node where the changes were made to all other nodes in the cluster. This step can be done either through the smitty interface (smitty cm_ver_and_sync.select) or the equivalent CLI command (/usr/es/sbin/cluster/utilities/clidare -rt -V 'normal'). After the verification is completed, ensure that the topology configuration on the other nodes in the cluster is identical to the node used for heartbeat network configuration changes.

**Deleting the device definitions from the server ODM**

To remove device definitions, use the C-SPOC interface to maintain consistency across all cluster nodes. The removal is done using the smitty interface as shown in Example 8-121.

**Example 8-121 Removing devices using C-SPOC interface “smitty cl_disk_man.rem.nodes”**

```
+--------------------------------------------------------------------------+
| Select Node(s) to Remove Disk From                                       |
|                                                                        |
| Move cursor to desired item and press F7.                               |
| ONE OR MORE items can be selected.                                      |
| Press Enter AFTER making all selections.                                |
|                                                                        |
| > itso_engine1                                                          |
| > itso_engine2                                                          |
|                                                                        |
| F1=Help                    F2=Refresh                  F3=Cancel              |
| F7=Select                  F8=Image                    F10=Exit               |
| Enter=Do                   /=Find                        n=Find Next            |
+--------------------------------------------------------------------------+

+--------------------------------------------------------------------------+
| Select A Disk to remove                                                  |
|                                                                        |
| Move cursor to desired item and press Enter.                             |
|                                                                        |
| 0004afca0b8dedbd ( hdisk2 on all cluster nodes )                         |
| 0004afca0b8e068a ( hdisk3 on all cluster nodes )                         |
| 0004afca0b8e136c ( hdisk4 on all cluster nodes )                         |
| 0004afca0b8e1f63 ( hdisk5 on all cluster nodes )                         |
|                                                                        |
| F1=Help                    F2=Refresh                  F3=Cancel              |
| F8=Image                    F10=Exit                   Enter=Do               |
| /=Find                        n=Find Next                      |
+--------------------------------------------------------------------------+
Remove a Disk From the Cluster

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[Entry Fields]

Nodes                                               drs_engine1, drs_engine2
Disk                                                0004afca0b8dedbd
KEEP definition in database                         no                     +

F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do

If you are not planning on reusing the devices, change the KEEP definition in database attribute to no to remove the ODM device entries.

Verifying that the device definitions are removed
Verify that all disk definitions from the old LUN set are now removed from the cluster nodes using `lsdev -Cc disk` and `mpio_get_config -Av` commands (Example 8-122).

Example 8-122 Results from issuing the `lsdev -Cc disk` and `mpio_get_config -Av` commands

```
# lsdev -Cc disk
hdisk0 Available 04-08-00-5,0 16 Bit LVD SCSI Disk Drive
hdisk1 Available 04-08-00-8,0 16 Bit LVD SCSI Disk Drive
hdisk6 Available 00-08-02     MPIO Other DS4K Array Disk
hdisk7 Available 00-08-02     MPIO Other DS4K Array Disk

# mpio_get_config -Av
Frame id 0:
  Storage Subsystem worldwide name: 60ab80026806c000049d5867c
  Controller count: 2
  Partition count: 1
  Partition 0:
    Storage Subsystem Name = 'DRS_FASTT1'
    hdisk  LUN #  Ownership          User Label
    hdisk6  11  B (preferred)      ITSO_vol_t001
    hdisk7  12  A (preferred)      ITSO_vol_t002
```
Unassigning old set of DS5000 LUNs from the cluster nodes

The DS5000 Storage Manager GUI window used for LUN mapping removal is shown in Figure 8-29.

The LUN mapping removal must be repeated for all the LUNs from the old LUN set. After you remove all the LUNs, the test migration is complete.

8.8.4 Migration Backout Considerations

The data migration process described in 8.8.3, “Detailed migration steps using PowerHA C-SPOC LVM mirroring” on page 355 removes the LV copy. After the LV copy is removed, you can no longer use that copy. The only accessible copy of the LV is on the new set of LUNs. If, for whatever reason, you decide to go back to the old set of LUNs, you need to repeat the migration process in the reverse direction. Depending on the amount of data, this process might take a significant amount of time. The downside of this method is that at the end of the migration, you have only one physical copy of the data. During the planning phase, before the migration starts, you need to decide whether the risk associated with having only one copy of data is acceptable.

The alternative approach allows you to preserve an extra copy of the data. The `splitlvcopy` command allows you to split a copy from the original LV and creates an LV from it. To maintain consistency of data, however, this command requires the application to be stopped and the file system to be unmounted. If you decide that having an extra copy is more important than doing the migration nondisruptively, use `splitlvcopy`.

The `splitlvcopy` command does not allow you to select which copy is used to form the new LV. It automatically uses the last copy of the LV. In the example, the command uses the copy on the new set of LUNs. If this is not the result you want, rename the original LV and then rename the new LV to use the original LV name. This way, the file system uses the copy on the new set of LUNs. If you need to go back to the original set of LUNs, all you need to do is to swap the LV names again. This process does not require any data to be copied.

---

**Figure 8-29**  Unassigning DS5000 LUNs from cluster nodes
8.9 Data migration using Linux LVM2 mirroring

Unlike Logical Volume Management (LVM) migration on other operating systems, Linux LVM2 can create a mirrored logical volume, but cannot split the mirror after it is synchronized.

Instead, Linux LVM can move physical extents in use by a volume group and its logical volumes to the new target devices using a block-by-block copy. This process uses an underlying mirror volume, but it does so on a segment basis and not an entire volume.

Restriction: This method has been presented as an “online” method (where application I/O can continue to run, while the migration is taking place) in some RedHat publications. However, testing with heavy I/O loads has found that migration can take a long time or stall altogether. Therefore, stop application I/O, or schedule the migration when access to the source volumes is light.

8.9.1 Introduction to Linux LVM2

Linux LVM is architecturally similar to other volume management facilities on other operating systems in terms of its storage virtualization and operating flexibility.

The Logical Volume Management on Linux is called LVM. The latest version of the LVM is called LVM2 and is compatible with earlier versions of Linux LVM1 except for certain clustering and snapshot features.

LVM2 requires the device mapper kernel driver, a generic device mapper that provides the framework for the volume management. For more information about Linux LVM2, see: http://tldp.org/HOWTO/LVM-HOWTO/

In this scenario, the system already has Linux LVM2 installed and the volumes to be migrated are part of an existing LVM volume group.

Linux LVM can move physical extents used by a volume group and its logical volumes to the new target devices using the pvmove command.

The pvmove command creates a temporary, mirrored logical volume and uses it to copy the data from the source device to the target device in segments. The original logical volume metadata is updated to use the temporary mirror segment until the source and target segments are in sync. It then breaks the mirror and the logical volume uses the target location for that segment. This process is repeated for each segment to be moved. After all segments are mirrored, the temporary mirror logical volume is removed and the volume group metadata is updated to use the new target volumes.

8.9.2 Multipathing considerations

The example uses the DS8000 Multipath Subsystem Device driver to provide multipathing and load balancing for the DS8000 LUNs on the host system. There is another alternative for multipathing with Linux called DM-Multipath or DM-MPIO. For more information about DM-MPIO and for a comparison between SDD and DM-MPIO, see “Considerations and Comparisons between IBM SDD for Linux and DM-MPIO” available at: http://www-1.ibm.com/support/docview.wss?uid=ssg1S7001664
8.9.3 Linux LVM migration scenario

The following environment and scenario are used to illustrate a Linux LVM migration.

**General outline of the Linux LVM migration**

The following are the high-level steps used to migrate data with Linux LVM. The source volumes are already in an LVM volume group.

1. Configure target storage volumes on the DS8000.
2. Configure the host to recognize the DS8000 volumes.
3. Install Multipath Subsystem Device Driver (SDD).
4. Modify LVM to recognize the SDD devices.
5. Partition the SDD vpath devices.
6. Configure lvm.conf to recognize SDD vpath devices.
7. Initialize SDD vpath devices to the LVM.
8. Extend the existing LVM volume group with the new vpath partitions.
9. Stop the application I/O and unmount the file systems. Use `pvmove` to migrate extents off the old device to the new device in the volume group.
10. Remove the old source devices from LVM.
11. Remove the old source devices from the system. Removing the devices is a disruptive process, and can be done at the next maintenance window.

**Migration environment**

The physical test environment is composed of the following components:

- Application Server: RedHat AS4, 2.6.9-42.ELsmp kernel
- Fibre Channel HBA: Qlogic qla2340 HBAs
- Source Storage Controller: EMC Symmetrix
- Source volumes:
  - `/dev/sdd1`: A primary partition on a 34-GB Symmetrix device
  - `/dev/sde1`: A primary partition on a 9-GB Symmetrix device
  - `/dev/sdf1`: A primary partition on a 9-GB Symmetrix device
- Target Storage Controller: IBM System Storage DS8300
- Target Volumes: A 36 GB and two 9 GB volumes are configured on the DS8000 to be used as targets for the migration.
- LVM version2
- LVM configuration:
  - One volume group
  - Two logical volumes on the volume group
  - Two file systems

*Important:* Back up the data on your source volumes before you start the migration process.

**LVM configuration before migration**

The current LVM configuration for the example consists of a single LVM volume group, called `migvg`, with two physical volumes with a single primary partition, `/dev/sdd1` and `/dev/sde1`.
Two logical volumes, /dev/migvg/medium_lv (34 G) and /dev/migvg/small_lv (8 GB), are built on migvg (Example 8-123). Note the total number of extents and free extents available on each physical volume. The extents will be migrated from the old physical volumes to the new physical volumes in the volume group.

Example 8-123  LVM Configuration before migration

```bash
[root@x345-tic-4 ]# vgdisplay -v migvg
    Using volume group(s) on command line
    Finding volume group "migvg"
    --- Volume group ---
    VG Name               migvg
    System ID
    Format                lvm2
    Metadata Areas        2
    Metadata Sequence No  37
    VG Access             read/write
    VG Status             resizable
    MAX LV                0
    Cur LV                2
    Open LV               0
    Max PV                0
    Cur PV                2
    Act PV                2
    VG Size               42.09 GB
    PE Size               32.00 MB
    Total PE              1347
    Alloc PE / Size       1344 / 42.00 GB
    Free PE / Size        3 / 96.00 MB
    VG UUID               edL624-ocV6-q128-QCT6-7kkC-34Q6-8twJsB
    --- Logical volume ---
    LV Name                /dev/migvg/medium_lv
    VG Name                migvg
    LV UUID                uHwA8J-4f2Y-RsXZ-4rHi-inyl-e3G2-y3LV06
    LV Write Access        read/write
    LV Status              available
    # open                 0
    LV Size                34.00 GB
    Current LE             1088
    Segments               2
    Allocation             inherit
    Read ahead sectors     0
    Block device           253:3
    --- Logical volume ---
    LV Name                /dev/migvg/small_lv
    VG Name                migvg
    LV UUID                TxMb0I-Ou9V-kGRX-1Mlu-kASE-32km-v1z8s3
    LV Write Access        read/write
    LV Status              available
    # open                 0
    LV Size                8.00 GB
    Current LE             256
    Segments               1
    Allocation             inherit
```
Configuring target storage on DS8000 using DSCLI

Physically cable the DS8000 to your SAN and zone each HBA port on your Linux host to the assigned ports from the DS8000. For best results, use no more than one host initiator per zone. However, more than one storage port can be a member of the zone. Ideally, you would have multiple zones with one host initiator, and one or more storage initiators in each zone for multiple logical paths. Multipath Subsystem Device Driver (SDD), which is installed later, handles the I/O and load balancing on each logical path.

Interactive mode DSCLI is used to configure the storage in this example. For more information about DSCLI, see DS8000 Command-Line Interface User’s Guide, SC26-7625-05. To configure the target storage, perform the following steps:

1. Create an empty DS8000 Volume Group to hold the DS8000 LUNs (Example 8-124).

Example 8-124 Create an empty volume group on the DS8000

dscli> mkvolgrp -hosttype LinuxRHEL x345-tic-4-vg
Date/Time: March 6, 2007 11:43:20 AM CET IBM DSCLI Version: 5.2.400.426 DS: IBM.2107-75L4741
CMUC00030I mkvolgrp: Volume group V0 successfully created.

2. Create fixed block DS8000 volumes and assign them to the Volume Group (Example 8-125).

Example 8-125 Create and verify volumes on the DS8000

dscli> mkfsvol -extpool p4 -cap 37 -type ds -name x345-tic-4_#h -volgrp v0 2000
Date/Time: March 6, 2007 12:16:31 PM CET IBM DSCLI Version: 5.2.400.426 DS: IBM.2107-75L4741
CMUC00025I mkfsvol: FB volume 2000 successfully created.

dscli> mkfsvol -extpool p4 -cap 10 -type ds -name x345-tic-4_#h -volgrp v0 2001
Date/Time: March 6, 2007 12:16:47 PM CET IBM DSCLI Version: 5.2.400.426 DS: IBM.2107-75L4741
CMUC00025I mkfsvol: FB volume 2001 successfully created.


dscli> lsfsvol -volgrp v0
Date/Time: March 6, 2007 12:18:36 PM CET IBM DSCLI Version: 5.2.400.426 DS: IBM.2107-75L4741
Name            ID   accstate datastate configstate deviceMTM datatype extpool cap (2^30B) cap (10^9B) cap (blocks)
===================================================================================================================
x345-tic-4_2000 2000 Online   Normal    Normal      2107-900  FB 512   P4             37.0           -     77594624
x345-tic-4_2001 2001 Online   Normal    Normal      2107-900  FB 512   P4             10.0           -     20971520
x345-tic-4_2002 2002 Online   Normal    Normal      2107-900  FB 512   P4             10.0           -     20971520

3. Create Host Definitions and assign the Volume Group to the host (Example 8-126).

Example 8-126 Create and verify the host on DS8000 using DSCLI

dscli> mkhostconnect -wwname 210000e08b879f35 -volgrp v0 -hosttype LinuxRHEL x345-tic-4-qla0
Date/Time: March 6, 2007 11:41:19 AM CET IBM DSCLI Version: 5.2.400.426 DS: IBM.2107-75L4741
Configuring the Linux host to recognize DS8000 volumes

Use procedures appropriate for your Linux Distribution and Host Bus Adapter Vendor to add device definitions for the new storage devices from the host system. This example uses the Qlogic dynamic reconfiguration utility to dynamically add the new DS8000 devices as shown in Example 8-127.

Example 8-127 Dynamic discovery of new DS8000 devices

[=root@x345-tic-4 ql-dynamic-tgt-lun-disc-2.2]# ./ql-dynamic-tgt-lun-disc.sh --scan
Please make sure there is no active I/O before running this script
Do you want to continue: (yes/no)? yes
Issuing LIP on host2
Scanning HOST: host2
......
Issuing LIP on host3
Scanning HOST: host3
......
Found
2:0:1:0
2:0:1:1
2:0:2:0
2:0:2:1
3:0:0:0
3:0:0:1
3:0:1:0
3:0:1:1

Installing System Storage Multipath Subsystem Device Driver (SDD)

SDD usually comes included with your DS8000 as part of the DS8000 Licensed Microcode Bundle. You can also obtain System Storage Multipath Subsystem Device Driver (SDD) for your host at:

http://www-01.ibm.com/support/docview.wss?rs=540&uid=ssg1S7001350

Install SDD on the Linux host using the Linux `rpm` command as shown in Example 8-128.

Example 8-128 Installing Multipath Subsystem Device Driver (SDD) on Linux

[root@x345-tic-4 tmp]# rpm -ivh IBMssdd-1.6.2.0-1.1686.rhel4.rpm
Preparing... 1:IBMssdd-1.6.2.0.1.1686.rhel4.rpm  [100%]
Added following line to /etc/inittab:
srv:345:respawn:/opt/IBMssdd/bin/sddsrv > /dev/null 2>&1
[root@x345-tic-4 tmp]# sdd start
Starting IBMsdd driver load: [ OK ]
Issuing killall sddsrv to trigger respawn... [ OK ]
Starting IBMsdd configuration: [ OK ]

List the vpath devices and their underlying sd devices using the \texttt{lsvpcfg} command (Example 8-129).

\textbf{Example 8-129} \hspace{1em} \textit{lsvpcfg} output

\begin{verbatim}
[root@x345-tic-4 tmp]# lsvpcfg
000 vpatha  ( 252,   0) 75L47412000 = 6005076305ff78600000000002000 = /dev/sdg /dev/sdi /dev/sdk /dev/sdm
001 vpathb  ( 252,  64) 75L47412001 = 6005076305ffc7860000000000002001 = /dev/sdh /dev/sdj /dev/sdl /dev/sdn
\end{verbatim}

For more information about SDD, see \textit{Multipath Subsystem Device Driver User's Guide}, SC30-4131-01.

\section*{Partitioning SDD vpath devices}

Partition your SDD vpath devices with your preferred partitioning utility. Remember that your target vpath partitions must be large enough to accommodate the extents from the source volumes. Set your partition type to 8e, Linux LVM.

In the example, \texttt{fdisk} was used to create a primary partition on each vpath and to set the partition type (Example 8-130).

\textbf{Example 8-130} \hspace{1em} \textit{Target vpath partitions}

\begin{verbatim}
[root@x345-tic-4 /]# fdisk -l /dev/vpatha
Disk /dev/vpatha: 39.7 GB, 39728447488 bytes
64 heads, 32 sectors/track, 37888 cylinders
Units = cylinders of 2048 * 512 = 1048576 bytes

    Device Boot Start  End   Blocks   Id  System
    /dev/vpatha1   1   37888 38797296   8e  Linux LVM

[root@x345-tic-4 /]# fdisk -l /dev/vpathb
Disk /dev/vpathb: 10.7 GB, 10737418240 bytes
64 heads, 32 sectors/track, 10240 cylinders
Units = cylinders of 2048 * 512 = 1048576 bytes

    Device Boot Start  End   Blocks   Id  System
    /dev/vpathb1   1  10240 10485744   8e  Linux LVM
\end{verbatim}

\section*{Modifying LVM configuration file to recognize SDD devices}

The \texttt{/etc/lvm/lvm.conf} file is the configuration file for LVM2. It is used during LVM initialization and LVM to recognize devices. To modify the \texttt{/etc/lvm/lvm.conf} file to recognize the SDD devices, perform the following steps:

1. Modify the filter section to accept the vpath devices, but not the underlying sd devices that make up the path (Example 8-131 on page 382). In the example, vpath devices, the sd devices that belong to the root logical volume, and the migration source devices are accepted. All other devices are rejected.
2. Add an entry for vpath in the types section. This entry adds vpaths as a block device type recognized by the LVM (Example 8-131).

```
Example 8-131 Filter and type settings in the /etc/lvm/lvm.conf file

filter = [ "a/vpath*/", "a/sda1/", "a/sda2/", "a/sdb1/", "a/sdd1/", "a/sde1/", "r/.*/" ]
... types = [ "vpath", 16 ]
```

For more information about the /etc/lvm/lvm.conf file, see the Linux man page on lvm.conf.

3. Run `vgscan` to allow the LVM to recognize the new disks (Example 8-132).

```
Example 8-132 vgscan

[root@x345-tic-4 ~]# vgscan
Reading all physical volumes. This may take a while...
Found volume group "migvg" using metadata type lvm2
Found volume group "VolGroup00" using metadata type lvm2
```

### Initializing the SDD vpath devices to the LVM

Initializing the SDD vpath devices prepares the disk for use by the LVM as a physical volume (PV). The `pvcreate` command initializes the device, in the example a disk partition, as shown in Example 8-133.

```
Example 8-133 Initialize vpath partitions for LVM

[root@x345-tic-4 ~]# pvcreate /dev/vpatha1
Physical volume "/dev/vpatha1" successfully created

[root@x345-tic-4 ~]# pvcreate /dev/vpathb1
Physical volume "/dev/vpathb1" successfully created
```

You can use a whole disk as an LVM physical volume. However, the LVM metadata written at the beginning of the disk will not be recognized by outside operating systems. It is therefore at risk of being overwritten.

The initialization process fails on a disk with an existing partition table on it. Double-check that you are using the correct disk for the LVM if you see this error.

**Tip:** You can use the following commands to delete the partition table on the disk if you want to use it for LVM. Use these commands with caution, as they will block access to the existing data on the disk.

```
# dd if=/dev/zero of=/dev/diskname bs=1k count=1
# blockdev --rereadpt /dev/diskname
```
Extending the existing LVM volume group with the new vpath partitions
The new vpath devices need to be added to the existing LVM volume group, migvg. Add the devices with the vgextend command as shown in Example 8-134.

Example 8-134  Using vgextend to add vpath devices to the volume group

```
[root@x345-tic-4 /]# vgextend migvg /dev/vpatha1
  Volume group "migvg" successfully extended
[root@x345-tic-4 /]# vgextend migvg /dev/vpathb1
  Volume group "migvg" successfully extended
```

Use the vgdisplay command to verify the addition of the new vpath devices (Example 8-135). Notice the free Physical Extents (PE) available to the migvg volume group on each new device.

Example 8-135  pvdisplay of the new vpath devices

```
[root@x345-tic-4 /]# pvdisplay /dev/vpatha1
--- Physical volume ---
  PV Name               /dev/vpatha1
  VG Name               migvg
  PV Size               36.97 GB / not usable 0
  Allocatable           yes
  PE Size (KByte)       32768
  Total PE              1183
  Free PE               1183
  Allocated PE          0
  PV UUID               QKFc5R-tGRt-wX85-IJeJ-o0r3-8T62-DtHuo1

[root@x345-tic-4 /]# pvdisplay /dev/vpathb1
--- Physical volume ---
  PV Name               /dev/vpathb1
  VG Name               migvg
  PV Size               9.97 GB / not usable 0
  Allocatable           yes
  PE Size (KByte)       32768
  Total PE              319
  Free PE               319
  Allocated PE          0
  PV UUID               AZweK9-jut4-T8fp-T9P8-7rLz-L2MQ-36958U
```

Using pvmove to migrate extents within the volume group
To use pvmove to migrate the extents within the VG, perform the following steps:

1. Stop all application I/O and sync your file systems. The file systems can remain mounted. However, to maintain consistency, make no new writes.

2. Start the migration of extents using the pvmove command. It migrates extents within a volume group from a source device to a target device as shown in Example 8-136 on page 384. Issue the command:

   pvmove source target

   Where source and target are your source and target devices in the volume group.

   Important: The pvmove command takes a long time to run because it performs a block-by-block copy.
Example 8-136  pvmove in progress

[root@x345-tic-4 ~]# pvmove /dev/sdd1 /dev/vpatha1
Detected pvmove in progress for /dev/sdd1
Ignoring remaining command line arguments
/dev/sdd1: Moved: 1.1%
/dev/sdd1: Moved: 1.3%
/dev/sdd1: Moved: 1.5%
/dev/sdd1: Moved: 1.6%
/dev/sdd1: Moved: 1.8%
/dev/sdd1: Moved: 1.9%
/dev/sdd1: Moved: 2.0%
/dev/sdd1: Moved: 2.2%
/dev/sdd1: Moved: 2.3%
/dev/sdd1: Moved: 2.5%
...

If pvmove needs to be stopped during the migration, the pvmove --abort command can be issued. The extents are spread between the old device and the new device. The migration can be continued by issuing a pvmove command without any parameters.

3. After pvmove is complete, verify the migration with the vgdisplay command as shown in Example 8-137. Notice that all the extents are free on /dev/sdd1 and are moved to /dev/vpatha1.

Example 8-137  vgdisplay showing the extents migrated from device sdd1 to vpatha1

[root@x345-tic-4 ~]# vgdisplay -v migvg
    Using volume group(s) on command line
    Finding volume group "migvg"
--- Volume group ---
  VG Name               migvg
  System ID
  Format                lvm2
  Metadata Areas        2
  Metadata Sequence No  4
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                2
  Open LV               2
  Max PV                0
  Cur PV                2
  Act PV                2
  VG Size               42.09 GB
  PE Size               32.00 MB
  Total PE              1347
  Alloc PE / Size       1344 / 42.00 GB
  Free PE / Size        3 / 96.00 MB
  VG UUID               edL624-ocV6-q128-QCT6-7kkC-34Q6-8twJsB

--- Logical volume ---
  LV Name                /dev/migvg/medium_lv
  VG Name                migvg
  LV UUID                Uwat6I-N7eL-zwCK-kGlV-0lRO-yh5x-X7qzv0
  LV Write Access        read/write
  LV Status              available
4. Repeat the `pvmove` command to move extents from /dev/sde1 to /dev/vpathb1. Continue with any additional devices with extents that need to be migrated.

5. Verify that the extents are now on the vpath devices using `vgdisplay` or `lvdisplay` commands as shown in Example 8-138.

Example 8-138  vgdisplay command showing physical volumes with all extents moved to vpath devices

--- Physical volumes ---

<table>
<thead>
<tr>
<th>PV Name</th>
<th>/dev/sdd1</th>
<th>PV UUID</th>
<th>54dWUW-0I1N-shHJC-wFTE-VCE2-xT4b-2sqZQx</th>
</tr>
</thead>
<tbody>
<tr>
<td>PV Status</td>
<td>allocatable</td>
<td>Total PE / Free PE</td>
<td>1078 / 1078</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PV Name</th>
<th>/dev/sde1</th>
<th>PV UUID</th>
<th>99jSJO-DZi1-29hN-PwR3-ovFA-gaON-OXDlAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PV Status</td>
<td>allocatable</td>
<td>Total PE / Free PE</td>
<td>269 / 3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PV Name</th>
<th>/dev/vpatha1</th>
<th>PV UUID</th>
<th>QKFc5R-tGRt-wX85-IJeJ-o0r3-8T62-DtHuol</th>
</tr>
</thead>
<tbody>
<tr>
<td>PV Status</td>
<td>allocatable</td>
<td>Total PE / Free PE</td>
<td>1183 / 105</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PV Name</th>
<th>/dev/vpathb1</th>
<th>PV UUID</th>
<th>AZweK9-jut4-T8fp-T9P8-7rLz-L2MQ-36958U</th>
</tr>
</thead>
<tbody>
<tr>
<td>PV Status</td>
<td>allocatable</td>
<td>Total PE / Free PE</td>
<td>319 / 319</td>
</tr>
</tbody>
</table>

--- Logical volume ---

<table>
<thead>
<tr>
<th>LV Name</th>
<th>/dev/migvg/small_lv</th>
</tr>
</thead>
<tbody>
<tr>
<td>VG Name</td>
<td>migvg</td>
</tr>
<tr>
<td>LV UUID</td>
<td>AEk2Bg-E3Yp-R5x6-tbyt-8Njv-z9ve-kah7qI</td>
</tr>
<tr>
<td>LV Write Access</td>
<td>read/write</td>
</tr>
<tr>
<td>LV Status</td>
<td>available</td>
</tr>
<tr>
<td># open</td>
<td>1</td>
</tr>
<tr>
<td>LV Size</td>
<td>34.00 GB</td>
</tr>
<tr>
<td>Current LE</td>
<td>1088</td>
</tr>
<tr>
<td>Segments</td>
<td>2</td>
</tr>
<tr>
<td>Allocation</td>
<td>inherit</td>
</tr>
<tr>
<td>Read ahead sectors</td>
<td>0</td>
</tr>
<tr>
<td>Block device</td>
<td>253:3</td>
</tr>
</tbody>
</table>

--- Physical volumes ---

<table>
<thead>
<tr>
<th>PV Name</th>
<th>/dev/sdd1</th>
<th>PV UUID</th>
<th>54dWUW-0I1N-shHJC-wFTE-VCE2-xT4b-2sqZQx</th>
</tr>
</thead>
<tbody>
<tr>
<td>PV Status</td>
<td>allocatable</td>
<td>Total PE / Free PE</td>
<td>1078 / 1078</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PV Name</th>
<th>/dev/sde1</th>
<th>PV UUID</th>
<th>99jSJO-DZi1-29hN-PwR3-ovFA-gaON-OXDlAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PV Status</td>
<td>allocatable</td>
<td>Total PE / Free PE</td>
<td>269 / 3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PV Name</th>
<th>/dev/vpatha1</th>
<th>PV UUID</th>
<th>QKFc5R-tGRt-wX85-IJeJ-o0r3-8T62-DtHuol</th>
</tr>
</thead>
<tbody>
<tr>
<td>PV Status</td>
<td>allocatable</td>
<td>Total PE / Free PE</td>
<td>1183 / 105</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PV Name</th>
<th>/dev/vpathb1</th>
<th>PV UUID</th>
<th>AZweK9-jut4-T8fp-T9P8-7rLz-L2MQ-36958U</th>
</tr>
</thead>
<tbody>
<tr>
<td>PV Status</td>
<td>allocatable</td>
<td>Total PE / Free PE</td>
<td>319 / 319</td>
</tr>
</tbody>
</table>
Removing the old source device from LVM

After all the physical extents are migrated, the physical volume can be removed from the volume group using the `vgreduce` command (Example 8-139).

**Example 8-139  vgreduce to remove a physical volume from a volume group**

```
[root@x345-tic-4 /]# vgreduce migvg /dev/sdd1
   Removed "/dev/sdd1" from volume group "migvg"
[root@x345-tic-4 /]# vgreduce migvg /dev/sde1
   Removed "/dev/sde1" from volume group "migvg"
```

Verify that the new vpath devices remain part of the volume group as shown in Example 8-140.

**Example 8-140  Verify migvg on vpath devices**

```
[root@x345-tic-4 /]# vgdisplay -v migvg
   Using volume group(s) on command line
     Finding volume group "migvg"
   --- Volume group ---
   VG Name               migvg
   System ID
   Format                lvm2
   Metadata Areas        2
   Metadata Sequence No  4
   VG Access             read/write
   VG Status             resizable
   MAX LV                0
   Cur LV                2
   Open LV               2
   Max PV                0
   Cur PV                2
   Act PV                2
   VG Size               42.09 GB
   PE Size               32.00 MB
   Total PE              1347
   Alloc PE / Size       1344 / 42.00 GB
   Free PE / Size        3 / 96.00 MB
   VG UUID               edL624-ocV6-q128-QCT6-7kkC-34Q6-8twJsB
```
--- Logical volume ---
LV Name /dev/migvg/medium_lv
VG Name migvg
LV UUID Uwat6I-N7eL-zwCK-kGlV-01R0-yh5x-X7qzv0
LV Write Access read/write
LV Status available
# open 1
LV Size 34.00 GB
Current LE 1088
Segments 2
Allocation inherit
Read ahead sectors 0
Block device 253:3

--- Logical volume ---
LV Name /dev/migvg/small_lv
VG Name migvg
LV UUID AEk2Bg-E3Yp-R5x6-tbyt-8Njv-z9ve-kah7qI
LV Write Access read/write
LV Status available
# open 1
LV Size 8.00 GB
Current LE 256
Segments 1
Allocation inherit
Read ahead sectors 0
Block device 253:4

--- Physical volumes ---
PV Name /dev/vpatha1
PV UUID tWVdi1-EpRZ-Nu5d-R27e-IhMq-AJgl-GqmSwa
PV Status allocatable
Total PE / Free PE 1183 / 105

PV Name /dev/vpathb1
PV UUID AZweK9-jut4-T8fp-T9P8-7rLz-L2MQ-36958U
PV Status allocatable
Total PE / Free PE 319 / 53

The logical volumes can now be remounted and application I/O started, or you can remove the old devices from the system.

Removing the old source device from the system
This operation can be done at the next available maintenance downtime because application I/O needs to be stopped and file systems unmounted.
1. Stop application I/O and unmount vpath devices.
2. Stop SDD. The underlying mapping between the DS8000 LUNs and the vpath devices will not be lost (Example 8-141).

Example 8-141 Stop SDD

```
[root@x345-tic-4 /]# sdd stop
Stopping IBMsd: [ OK ]
```
3. Change SAN zoning to remove access to old source volumes. This change preserves the volumes on the old storage controller if they need to be restored.

4. Use procedures appropriate for your operating system and Host Bus Adapter Vendor to remove device definitions for the old storage controller devices from the host system.

The example uses the Qlogic dynamic reconfiguration utility to dynamically remove the old storage controller devices as shown in Example 8-142.

Example 8-142  Qlogic dynamic reconfiguration tool to remove old storage devices

```
[root@x345-tic-4 ql-dynamic-tgt-lun-disc-2.2]# ./ql-dynamic-tgt-lun-disc.sh -r -s
Please make sure there is no active I/O before running this script
Do you want to continue: (yes/no)? yes
Issuing LIP on host2
Scanning HOST: host2
............
Issuing LIP on host3
Scanning HOST: host3
............
Removed
 2:0:0:0
 2:0:0:14
 2:0:0:23
 2:0:0:3
```

5. Start SDD (Example 8-143).

Example 8-143  Start SDD

```
[root@x345-tic-4 ~]# sdd start
Starting IBMsd driver load: [ OK ]
Issuing killall sddsrv to trigger respawn...
Starting IBMsd configuration: [ OK ]
```

6. Remount LVM logical volumes.

7. Verify SDD configuration using `datapath` commands as shown in Example 8-144.

Example 8-144  Datapath query device command output

```
[root@x345-tic-4 ~]# datapath query device
Total Devices : 3

DEV#:  0  DEVICE NAME: vpatha  TYPE: 2107900  POLICY: Optimized Sequential
       SERIAL: 75L47412000
============================================================================
Path#  Adapter/Hard Disk       State     Mode          Select     Errors
0      Host2Channel10/sdg      OPEN   NORMAL               1          0
1      Host2Channel10/sdp      OPEN   NORMAL               0          0
2      Host3Channel10/sdi      OPEN   NORMAL              40          0
3      Host3Channel10/sdl      OPEN   NORMAL               0          0

DEV#:  1  DEVICE NAME: vpathb  TYPE: 2107900  POLICY: Optimized Sequential
       SERIAL: 75L47412001
============================================================================
```
8. Restart application I/O.

8.10 Data migration using network block devices

Network block devices are block devices where the real device is attached to a different Linux host. The device accepts normal block commands to write, read, and maintain data. However, the driver forwards the I/O through a TCP network to the server that the actual device is connected to. For migrations, the network block device can be used as transport mechanism and then discarded when the migration is done.

Using this technique, it is possible to migrate data, block by block, to another destination. The easiest way to migrate data to the new device is using shell copy commands like `cp`, `dd`, or `cpio`. A more elegant way is to establish a mirror using the network devices and allow the background synchronization copy the data to the new target.

The following example involves NDB and DRDB, which are both integrated in the Linux kernel. Other implementations are also referenced.

8.10.1 General architecture

The Linux Network Block Device `nbd`, is a basic implementation of a generic network block device that is included in the kernel since version 2.1.101. It provides read-only and read/write access to a remote device. In read-only mode, many clients can mount the remote device. For write access, only one host is allowed to access the remote device at a time. There is no access control included in the code. Therefore, the data on the network device can be deleted if more than one client tries to write to this device.
Figure 8-30 shows the architecture in which the migration using nbd will work. At the target server, a nbd server is started for the device /dev/sdb1. The source server accesses this device using the nbd-client. The new device is named /dev/nbd0.

The migration itself is operated at the source location.

![Diagram of migration architecture for network block devices]

When the nbd server is started, it listens to a TCP or UDP port assigned by the administrator. The server is started with the command shown in Example 8-145.

**Example 8-145  Starting the nbd-server**

```bash
# nbd-server 5000 /dev/sdb1
# _
```

In this example, the nbd device is represented by a file in the directory /export. When the server starts, it listens at the port 5000 for incoming requests.

After the server is started, the client can create a block device by referencing the IP address of the server and the port where the server listens (Example 8-146).

**Example 8-146  Create a nbd device**

```bash
# nbd-client 9.155.33.67 5000 /dev/nbd0
# _
```

The network block device is now ready to be used. For example, create a file system, mount it and allocate files and directories. For data migrations, however, other methods are of interests as shown in the following chapters.
8.10.2 Using shell commands

When the network block device is created, a file system can be allocated. Data can be migrated, file by file, with `cp` or `cp -R` shell commands. You can also copy the whole device in a single `dd` statement to the network device as shown in Example 8-147.

Example 8-147 Copying a raw device to the nbd

```
# dd if=/dev/sda2 of=/dev/nbd0 bs=1024
# _
```

This method requires that the applications using the data on the disk `/dev/sda2` must be shut down. In addition, the file system must be unmounted to catch the recent I/O from the cache. Depending on the used capacity of the disk and the bandwidth of the network, this process might take a while before all data is transmitted.

8.10.3 Using software RAID1

The easiest way of replicating the data using network devices is to establish RAID1 as a software RAID using the tool `mdadm`. The `mdadm` tool can provide different RAID levels by using real block devices. For data migration, RAID1 is mirroring between two volumes. After the RAID1 is created, both volumes are synchronized in the background. When the synchronization is completed, all updates by applications or users are replicated to both devices. When the migration is finalized, close the applications, unmount the file systems, and close the RAID1 using the `mdadm` tool.

To use software RAID1 to migrate data, perform the following steps:

1. Establish the RAID1 device (Example 8-148).

```
Example 8-148 Create the RAID1 out of the local disk and the nbd0 device

# mdadm --create /dev/md0 --level=mirror --raid-devices=2 /dev/sda2 /dev/nbd0
mdadm: /dev/sda2 appears to contain an ext2fs file system
    size=1048576K  mtime=Thu Jul  7 14:30:34 2011
Continue creating array? y
mdadm: array /dev/md0 started.
#
```

2. Close all applications and mount the file systems.

3. A new device is allocated that writes the data to both devices. Mount the file system to this device.

4. Transmit the device `/dev/sda2` to the network block device `/dev/nbd0`. The result is a new device, `/dev/md0`, which is now the RAID1 device. The synchronization of both mirrors starts immediately with this command. The progress of the synchronization can be monitored with the command shown in Example 8-149.

```
Example 8-149 Query the current status of the synchronization

# cat /proc/mdstat
Personalities : [raid1]
md0 : active raid1 nbd0[1] dm-3[0]
    1048512 blocks [2/2] [UU]
    [==========>..................] resync = 31.2% (327680/1048512) finish=0.5min
    speed=23405K/sec
```
5. You can see a progress bar which shows the status of the synchronization. When the synchronization is done, you see the status shown in Example 8-150.

Example 8-150 Status, when synchronization is done

```
# cat /proc/mdstat
Personalities : [raid1]
md0 : active raid1 nbd0[1] dm-3[0]
       1048512 blocks [2/2] [UU]
unused devices: <none>
# _
```

6. The file system for the applications can be mounted at any time after the RAID1 was established (Example 8-151).

Example 8-151 Mount the RAID1 device

```
# mount /dev/md0 /mnt
# df
Filesystem 1K-blocks Used Available Use% Mounted on
/dev/sda1 14761044 8922208 5089004  64% /
none 4071508 412 4071096 1% /dev
none 4076096 1932 4074164 1% /dev/shm
none 4076096 188 4075908 1% /var/run
none 4076096 0 4076096 0% /var/lock
none 4076096 0 4076096 0% /lib/init/rw
/dev/md0 1032088 103468 876192 11% /mnt  <===
# _
```

7. Any updates from users or the applications are now mirrored to both underlaying devices. When you are ready the cut over, close the applications and unmount the file systems, then close the RAID1 as shown in Example 8-152.

Example 8-152 Cut over to the target device

```
# umount /mnt
#
# mdadm --stop /dev/md0
mdadm: stopped /dev/md0
#
# cat /proc/mdstat
Personalities : [raid1]
unused devices: <none>
# _
```

8. When the RAID1 was established, a md-superblock was written to each device. To mount a remote device on the target system, this superblock must be removed using the command shown in Example 8-153.

Example 8-153 Cleanup the superblock at the target system

```
# mdadm --zero-superblock /dev/nbd0
#
```
The migration is now complete. The remaining operations are to stop the nbd-server at the target system, stop the nbd-client at the source system, and clean up the RAID1.

### 8.10.4 Summary

The Network Block Device (nbd) can be used for a basic migration to new Linux systems and new storage devices. For migrations involving only a few disks, this type of migration can easily be implemented.

However, two major issues need to be considered first:

- The communication between the server and the client is realized by specifying a TCP user port. This communication can be a security issue in some environments.
- There is no control mechanism that prevents more than one client from writing data to the server. The lack of a control mechanism is a high risk to corrupt the data on the server. You need to make sure that no other client has access to the server device.

### 8.10.5 Using Linux LVM2 mirroring

Another method to migrate data to a network block device is using the Linux Logical Volume Manager LVM. The source data devices must already be part of an existing LVM configuration. Otherwise, the devices themselves must first be migrated to LVM.

The volume group is then extended by the new nbd device. Transmit the data using the methods as described in 8.9, “Data migration using Linux LVM2 mirroring” on page 376.
Using TDMF for z/OS

This chapter describes the TDMF for z/OS product. It addresses the steps required to use the product in a typical local migration scenario.

This chapter contains the following sections:

- TDMF for z/OS overview
- Customer scenarios
- TDMF preferred practices
9.1 TDMF for z/OS overview

Transparent Data Migration Facility (TDMF) for z/OS is a host-based software solution for data migration in a z/OS environment. TDMF offers the following functions:

- Is user-initiated and controlled
- Allows for full system sharing throughout the data center
- Guarantees full access to the data at any point during a migration operation
- Supports dynamic takeover on the part of the target device

9.1.1 Data migration and migration tool definitions and characteristics

Working with TDMF for z/OS includes the following characteristics:

- Multiple volume migrations can be established during a single session.
- The migration tool is dynamically activated and terminated.
- Applications remain unaware that migration is underway. The data on the source device is continuously and fully accessible for read and write activity.
- After the synchronization of source and target volumes is complete, the takeover of the target device is nondisruptive.
- The tool supports a multiple-system shared data environment.
- The tool guarantees complete physical data integrity.
- The use of the tool is not restricted to any control unit model type or device type. Except as noted, all devices that are online in the data center can participate in a migration session as required.
- All volumes (source and target) of a migration must be online.
- No data set can be allocated to a target volume during migration.

Restriction: TDMF for z/OS does not allow access to the target volume during the migration process.

- A source volume might not contain an active local page data set or swap data set.
- The source and target volumes must be of the same track geometry.

These characteristics represent the ideal of a transparent and non-disruptive migration facility.

9.1.2 Terminology

These terms describe the TDMF installation and migration process:

- Master system: The TDMF system, running as an z/OS batch job or started task that is responsible for the data copy function. There is only one master system in a TDMF session.
- Agent system: An associated TDMF z/OS image running in a shared storage environment with the master. To ensure data integrity, any z/OS LPAR with access to the volumes must be run on either the master or an agent system. The master and associated agent systems communicate through a shared system communications data set (COMMDS).
- Source volume: The DASD volume containing the data to be migrated.
- Target volume: The DASD volume receiving the migrated data.
- Migration pair: The source and target volumes for a single migration.
- Migration group: A group of volume pairs using the same group name.
- Migration session: A master system plus any agents, if applicable, and pairs or groups to be processed in a single TDMF execution.
- Synchronization: The period when the following steps occur:
  a. I/O is inhibited to the source devices (quiesce phase).
  b. The last of the updates are collected and reflected to the target devices (synchronization phase).
  c. The source and target volume serial numbers are exchanged (redirect phase).
  d. The I/O is resumed to the new source (original target) device (resume phase)
- Swap migration: A session in which the source and target VOLSERs are swapped and I/O redirection to the new source occurs at volume termination. The original source device is marked offline.

### 9.1.3 TDMF z/OS components architecture

Figure 9-1 depicts the TDMF z/OS architecture and components. Volumes can be moved between different storage vendors or inside the same storage unit.

The architecture has the following components:

- TDMF TSO monitor: The TDMF TSO monitor is used to view active or past migrations. It is installed at the same time as the TDMF tool. This feature consists of REXX execs, which require ISPF version 3.6 and TSO/E Version 2.4 at minimum.
Communications data set: The communications data set or COMMDS is used to pass information between systems participating in a TDMF session. This data set contains the status and messages related to a specific session. The COMMDS also serves as the input file to the TDMF TSO monitor.

Master and agents: Each LPAR that has access to the storage subsystem containing the source and the target volumes must run an agent. Each master or agent within a session must have access to the same COMMDS. Multiple agent systems can be involved in a session locally.

**Important:** Because of a possible data integrity exposure, all systems accessing migration volumes must be identified to the master system. TDMF includes various controls and checks so you do not make the following errors:

- Assign or direct conflicting migrations to the same devices
- Attempt migrations to nonexistent devices
- Use the same COMMDS for two simultaneous or overlapping migration sessions

For audit purpose, do not use the same COMMDS for different sessions.

**Master system responsibilities**
The TDMF Master system is responsible for the following tasks:

- Initialize the TDMF Master system environment and the COMMDS
- Start and control each session for all participating systems
- Monitor source volume user I/O activity to detect updates
- Monitor target volume user I/O activity to prevent updates
- Copy data from the source volume to the target volume
- Process detected source volume updates from all systems
- Perform refresh operations to the target volume to reflect the update activity on the source volume
- Check the internal health of the master environment and the health of all agent systems

**Agent system responsibilities**
The TDMF Agent system is responsible for the following tasks:

- Initialize the TDMF Agent environment and establish communications to the master system through the COMMDS
- Acknowledge and process migration requests (volumes to be moved) from the master system
- Monitor source volume user I/O activity and detect updates
- Monitor target volume user I/O activity to prevent updates
- Notify the master system of source volume update activity through the COMMDS
- Check the internal health of the agent environments and the health of the master system

**9.1.4 TDMF z/OS process flow**

The TDMF z/OS process is a phased approach to make sure that data integrity is maintained in each phase. Data migration sessions can be interrupted at any time with no data loss. During the migration progress, all updates are written to the original source volume and are replicated by TDMF to the target volume. If an I/O violation occurs, the session is terminated.
TDMF system initialization process

Only after successful initialization of all systems in a TDMF session does migration proceed. If any violation occurs during system initialization on any system defined in the session, no migrations are performed.

If all systems in the session are not started within a 15-minute interval, the session does not complete system initialization. If you start a system that is not part of an active session, TDMF terminates the master job and all agent jobs using the same COMMDS.

If you use System Authorization Facility (SAF), and any volume involved in the migration session fails SAF, the migration session itself fails.

SAF requirements are:

- Swap type migrations require ALTER authority on the source and target volumes.
- Point-in-Time migrations require READ authority for the source volume and UPDATE authority for the target volume.

Volumes in a session can be terminated using the TDMF TSO Monitor or Batch Monitor on the Master system before initialization of the agent systems. If you select the History option to automatically record information about the migration session, the recording requires UPDATE authority for the data set. For more information, see the System Authorization Facility (SAF) and TDMF Installation and Reference, TDM-Z53IR.PDF at:


The master system initiates and controls all migrations. The agents must each phase to proceed. If any system detects a violation, that specific migration terminates. Depending on the state of the current migration, you must perform back-out processing to establish the original status before the migration session started.
TDMF migration phases

The TDMF migration process flow can be broken into major phases as shown in Figure 9-2.

- Initialization phase: During this phase, all participating systems confirm the validity of the source and target volumes. The phrase includes the following activities:
  - Volume acknowledgement: Volumes that require an acknowledgement are not eligible for confirmation and selection until one is received. This acknowledgement can use the TDMF TSO Monitor, a Batch Monitor, or the IBM MVS™ Write-to-Operator/Write-to-Operator with Reply (WTO/WTOR).
    
    This acknowledgement is required for volumes being migrated whose disaster recovery or mirror techniques are not compatible unless the ALLOWmirrorchange(NOACKnowledge) option is specified. TDMF recognizes volumes that use Peer-to-Peer Remote Copy (PPRC), extended remote copy (XRC), TrueCopy from Hitachi, and Symmetrix Remote Data Facility (SRDF). These volumes are recognized with or without Consistency Groups from EMC.

    The disaster recovery or mirroring type must be compatible between the source and target volumes. Compatibility includes the vendor command interface structure and parameters. If they are not compatible, the ALLOWmirrorchange option must be specified and an acknowledgement received.

  - Volume confirmation: Volumes that require confirmation are not eligible for volume or group selection until a confirmation is received. This confirmation can use the TDMF TSO Monitor, a Batch Monitor, or the MVS Write-to-Operator/Write-to-Operator with Reply (WTO/WTOR). The order of confirmation determines the order of volume selection. Volumes or groups that do not require confirmation are immediately available for volume or group selection.
– Volume selection: By default all volume pairs defined in a session are automatically selected during the initialization phase. However, volume selection is affected when certain user options are specified. The options that affect volume selection are volume confirmation, number of concurrent volumes, active in copy, group options, and allow mirroring changes.

– Volume initialization: Initialization of all volume level control blocks and page fixing of all real storage frames are necessary for a volume migration.

- Activation phase: In this phase, the I/O monitor is activated by TDMF for the source and target device. All systems in the session then attempt to allocate each source and target volume, preventing them from being inadvertently varied offline.

- Copy phase: In this phase, the master system begins a COPY volume task to copy data from the source volume to the target volume. Each source volume has an independent COPY volume task in the migration session.

If there are source volume updates during the copy phase, the master system collects the updated information to be processed during the refresh phase. Changes to the source volume are kept in a bitmap located in the COMMDS.

- Refresh phase: When the COPY volume phase completes one pass of the source volume, the target volume receives the updates made to the source volume. This phase is controlled by the master system.

Multiple refresh phases occur until TDMF determines that synchronization of the target volume can be achieved, at which time the master system quiesces the source volume.

- Quiesce phase: In this phase, the master system stops all I/O activity to the source volume. When an agent system receives a quiesce request, the agent system sends the master system the final group of detected updates.

- Synchronize phase: When all the systems comply with the quiesce request, all detected updates are written to the target volume.

- Volume I/O redirect phase: In this phase, TDMF redirects I/O activity from the source volume to the target device. The master system requests that all systems do a redirect, and confirms that these redirects are successful. When this process is complete, the master system rewrites the volume labels to change the serial numbers and performs redirect processing.

**Tip:** If an LPAR in the same TDMF session failed an I/O redirection, TDMF resets the redirection on all LPARs. TDMF goes on to the resume phase with an I/O error.

- Resume phase: Immediately after successful I/O redirect processing, the master system performs RESUME processing and initiates the RESUME request for the agent systems. Resuming allows user I/O to continue to the volume on its new device. After all systems process the RESUME request, the original (source) device is marked offline.

If an I/O error during the quiesce, synchronize, or I/O redirection phases, the source device is re-enabled for I/O and the UCB quiesce is removed.

- Terminate phase: When a volume completes a migration, the fixed storage of that volume is released for possible reuse in the current session. All dynamic allocations are also removed.

### 9.1.5 TDMF z/OS hardware compatibility

TDMF is designed to support any Count Key Data/Extended (CKD/E) capable control units as a source or a target storage subsystem. These functions include Hyper-Volumes,
Flexible-Volumes, and all user-defined volume sizes. Handling of Extended Address Volumes (EAVs) depends on the z/OS level. See the TDMF Reference Manual for more details.

Because TDMF is a host software migration tool, there are no hardware prerequisites.

9.1.6 Installing TDMF z/OS

There are two ways to install TDMF on a system:

- With SMP/E for z/OS
- Without SMP/E (preferred for migration service)

Pre-Installation considerations

The master system requires additional storage, depending on the number of volumes in the session. This additional storage is not a significant amount, and is only allocated for volumes with the Parallel Access Volumes (PAV) option set:

- 10 or less volumes = 4 K
- 10 - 32 volumes = 4 K to 12 K
- 32 - 64 volumes = 12 K to 24 K
- 64 - 128 volumes = 24 K to 48 K
- 128 - 256 volumes = 48 K to 96 K
- 256 - 512 volumes = 96 K to 192 K

The Softek TDMF for z/OS Installation Assurance Guide for V5R3.0, TDM-Z53IA-001.pdf, has complete pre-installation and post-installation checklists that need to be followed. Be sure to read the copy of this document that matches the level of software you are installing. It is available at:


Installing with SMP/E

An installation standard procedure must be provided to indicate to SMP/E where to locate all the required data sets. Unless predefined within the consolidated software inventory (CSI) using DDDEFs, allocate all the DDnames for the Softek TDMF target and distribution libraries.

The following members in the distributed SAMPLIB can be copied and tailored to install Softek TDMF. The DLIBZONE and TARGZONE must be updated in the samples to reflect the zone definitions for the site.

The order of installation is:

1. ALLCSMP creates the SMP/E files LOG, LOGA, MTS, PTS, SCDS, and STS.
2. INITCSI creates and initialize the CSI, global, target, and distribution zones.
3. DDDEF creates the DDDEF entries in SMP/E.
4. SMPE is the sample procedure that the following jobs run.
5. SMPERECE performs an SMP/E RECEIVE of the Softek TDMF product.
6. SMPEAPK performs an SMP/E APPLY CHECK of the Softek TDMF product.
7. SMPEAPP performs an SMP/E APPLY of the Softek TDMF product.
8. SMPEACK performs an SMP/E ACCEPT CHECK of the Softek TDMF product.
9. SMPEACC performs an SMP/E ACCEPT of the Softek TDMF product.

The following procedure creates a complete and separate SMP/E environment for Softek TDMF. Alternatively, you can install the product in any other SMP/E structure, However, you must edit the jobs to fit your environment.
To install Softek TDMF using SMP/E, perform the following steps:

1. Edit ALLCSMP with the TDMFEDIT exec and submit (allocates MTS, PTS, SCDS, STS, LOG, and LOGA data sets).
2. Edit SMPE with the TDMFEDIT exec and copy to PROCLIB.

   **Note:** In JES3 environments, you might need to separate this step into multiple jobs.

3. Edit INITCSI with the TDMFEDIT exec and submit (calls SMPE).
4. Edit DDDEF with the TDMFEDIT exec and submit.
5. Edit SMPEREC with the TDMFEDIT exec and submit.
6. Edit SMPEAPK with the TDMFEDIT exec and submit.
7. Edit SMPEAPP with the TDMFEDIT exec and submit.
8. Edit SMPEACK with the TDMFEDIT exec and submit.
9. Edit SMPEACC with the TDMFEDIT exec and submit.

### Installing without SMP/E

For data migration purposes, it is easier to install without SMP/E. In this case, the INSTALL member in the sample library performs an IEBCOPY upload of the modules. TDMF maintenance is not provided, and you must therefore install the full package again. For more information, see *TDMF Installation and Reference (Maintenance Overview)*.

   **Note:** If the installation has a security package on the z/OS system on which TDMF is installed, you must modify the security package. Otherwise TDMF will not run properly.

### 9.1.7 TDMF z/OS customer requirements

TDMF supports all z/OS-based operating systems that are currently supported by IBM. This product is not suitable for use with IBM z/VM® and IBM VM/ESA®, VM/XA, VSE/ESA, VSE/XA, or DOS. The Operating Systems Support Matrix is at the following web address:


If a z/OS system is available, copy z/VM, Linux for System z, and IBM z/VSE® volumes (3390-x volumes only) using REPLICATE instead of MIGRATE. These operating systems must be OFFLINE because there is no AGENT available. Define the source and target volumes to the z/OS system and run the same jobs as in z/OS, but with parameter REPLICATE instead of MIGRATE. After the data is replicated, use ICKDSF REFORMAT to rename the target volumes to their original VOLIDs.

Special considerations must be taken when z/OS is running under z/VM when allocating the COMMDS. For more information, see the TDMF installation manual at:

Security

If the installation has a security package on the z/OS system on which TDMF is installed, you need to modify the security package. Otherwise TDMF does not run properly. Check the profiles and command tables.

Limiting access to the TDMF authorized library to prevent unauthorized use of the TDMF system can be accomplished through security packages. The user for the SYSOPTN batch job (provided by the TDMF installation) must have UPDATE authority for the library indicated by the SECCOM DD statement. The master migration job also needs UPDATE authority for the SECCOM file if the authorization key specifies volume or terabyte limits. To update authorization keys using Option 10 of the TDMF TSO Monitor as shown in Figure 9-3, you must have UPDATE authority for the TDMLLIB library. In addition, the library must be allocated as SECCOM in the SYSOPTN job.

Tip: Perform periodic checks of the Required IBM Maintenance, IBM Technical Support (PTFs), and Technical Information Bulletins (TIBs). These requirements must be implemented to ensure successful TDMF operation.

Required IBM Maintenance:

IBM Technical Support:

Technical Information Bulletins:

Figure 9-3  TDMF license keys

If the History option is selected, UPDATE authority is required for the data set specified. When viewing the history file (and any COMMDS) using the TDMF TSO Monitor, you must have READ authority.
System Authorization Facility
If you want to use the System Authorization Facility (SAF), select VOLUME SECURITY = YES when using the SYSOPTN batch job in SAMPLIB. For more information about the TDMF system defaults, see TDMF System Defaults in the TDMF Installation Manual at:

For a swap migration, ALTER authority must be in effect for the source and target volumes. Error messages are issued for all volumes not meeting these requirements in a session.

TDMF checks for two types of classes:
> CLASS=DATASET for the COMMDS history data set and the TDMF load library defined on the SECCOM DD statement
> CLASS=DASDVOL for volumes allowed in a pairing

If the authorization keys specify volume or terabyte limits, the TDMLLIB (SECCOM DD statement) must have update authority for the user ID submitting the jobs.

9.2 Customer scenarios

This section illustrates a simple example of migrating data from an ESS 800 to a DS8000. In the example four volumes are moved, but the process would be the same for any number of volumes.

9.2.1 Test environment

The test environment has the following characteristics:
> Hardware and software setup as shown in Figure 9-4

![Example: Customer scenario](image)
Data Migration to IBM Disk Storage Systems

- z/OS level 01.12.00 (Figure 9-5).

IEE421I RO *ALL,D IPLINFO 955
MCECEBC RESPONSES --------------------------------------
IEE254I 18.27.15 IPLINFO DISPLAY 954
SYSTEM IPLED AT 12.41.01 ON 01/21/2011
RELEASE z/OS 01.12.00 LICENSE = z/OS
USED LOADS9 IN SYS1.IPLPARM ON CA13
ARCHLVL = 2 MTLSHARE = N
IEASYM LIST = BC
IEASYS LIST = 00
IODF DEVICE: ORIGINAL(CA13) CURRENT(CA13)
IPL DEVICE: ORIGINAL(CA10) CURRENT(CA10) VOLUME(CEBCR1)
MZBCVS2 RESPONSES --------------------------------------
IEE254I 18.27.15 IPLINFO DISPLAY 855
SYSTEM IPLED AT 13.35.54 ON 01/27/2011
RELEASE z/OS 01.12.00 LICENSE = z/OS
USED LOADS9 IN SYS1.IPLPARM ON CA13
ARCHLVL = 2 MTLSHARE = N
IEASYM LIST = S2
IEASYS LIST = 00
IODF DEVICE: ORIGINAL(CA13) CURRENT(CA13)
IPL DEVICE: ORIGINAL(CA10) CURRENT(CA10) VOLUME(CEBCR1)

Figure 9-5  Display IPL information including the z/OS level
Two LPARs (LPAR 4 and 5) as seen in Figure 9-6

<table>
<thead>
<tr>
<th>ID</th>
<th>CPU</th>
<th>SERIAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>+</td>
<td>04423A2064</td>
</tr>
<tr>
<td>01</td>
<td>+</td>
<td>14423A2064</td>
</tr>
<tr>
<td>02</td>
<td>+</td>
<td>24423A2064</td>
</tr>
<tr>
<td>03</td>
<td>+</td>
<td>34423A2064</td>
</tr>
<tr>
<td>04</td>
<td>+</td>
<td>44423A2064</td>
</tr>
<tr>
<td>05</td>
<td>+</td>
<td>54423A2064</td>
</tr>
<tr>
<td>06</td>
<td>+</td>
<td>64423A2064</td>
</tr>
</tbody>
</table>

CPC ND = 002064.116.IBM.51.00000002423A
CPC SI = 2064.116.IBM.51.000000000002423A
CPC ID = 00
CPC NAME = P002423A
LP NAME = BVS1       LP ID = 4
**MIF ID = 4**

<table>
<thead>
<tr>
<th>ID</th>
<th>CPU</th>
<th>SERIAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>+</td>
<td>05423A2064</td>
</tr>
<tr>
<td>01</td>
<td>+</td>
<td>15423A2064</td>
</tr>
<tr>
<td>02</td>
<td>+</td>
<td>25423A2064</td>
</tr>
<tr>
<td>03</td>
<td>+</td>
<td>35423A2064</td>
</tr>
<tr>
<td>04</td>
<td>+</td>
<td>45423A2064</td>
</tr>
<tr>
<td>05</td>
<td>+</td>
<td>55423A2064</td>
</tr>
<tr>
<td>06</td>
<td>+</td>
<td>65423A2064</td>
</tr>
</tbody>
</table>

CPC ND = 002064.116.IBM.51.00000002423A
CPC SI = 2064.116.IBM.51.000000000002423A
CPC ID = 00
CPC NAME = P002423A
LP NAME = BVS2       LP ID = 5
**MIF ID = 5**

**Figure 9-6  Displaying processor information**

- DISK subsystems: IBM TotalStorage ESS 800 and IBM System Storage DS8000
- Channel attachment (ESCON / FICON) for storage
9.2.2 Migration steps

To migrate data using TDMF for z/OS, perform the following steps:

1. Verify that all volumes to be migrated are online to all LPARs involved in the migration as shown in Figure 9-7. Send a display command for the source devices to all LPARs to query the status and volser. In the example, one volume is offline on one of the LPARs.

```
RO *ALL,D U,,,6000,4
IEE421I RO *ALL,D U,,,6000,4 112
MCECEBC RESPONSES ------------------------------
IEE457I 22.56.27 UNIT STATUS 111
UNIT TYPE STATUS VOLSER VOLSTATE
6000 3390 A RS6000 PRIV/RSDNT
6001 3390 O RS6001 PRIV/RSDNT
6002 3390 A RS6002 PRIV/RSDNT
6003 3390 O RS6003 PRIV/RSDNT
MZBCVS2 RESPONSES ------------------------------
IEE457I 22.56.27 UNIT STATUS 909
UNIT TYPE STATUS VOLSER VOLSTATE
6000 3390 O RS6000 PRIV/RSDNT
6001 3390 O RS6001 PRIV/RSDNT
6002 3390 O RS6002 PRIV/RSDNT
6003 3390 OFFLINE RS6003 PRIV/RSDNT
```

Figure 9-7 Display source device address at all LPARs

2. Bring the volume online by issuing a VARY ONLINE command to all LPARs for all devices in the range (6100-6103) as shown in Figure 9-8.

```
RO *ALL,VARY 6000-6003,ONLINE
IEE421I RO *ALL,VARY 6000-6003,ONLINE 163
MCECEBC RESPONSES ------------------------------
IEE457I 22.58.30 UNIT STATUS 162
UNIT TYPE STATUS VOLSER VOLSTATE
6000 3390 0 RS6100 PRIV/RSDNT
6001 3390 0 RS6101 PRIV/RSDNT
6002 3390 0 RS6102 PRIV/RSDNT
6003 3390 0 RS6103 PRIV/RSDNT
MZBCVS2 RESPONSES ------------------------------
IEE457I 22.58.30 UNIT STATUS 482
UNIT TYPE STATUS VOLSER VOLSTATE
6000 3390 0 RS6100 PRIV/RSDNT
6001 3390 0 RS6101 PRIV/RSDNT
6002 3390 0 RS6102 PRIV/RSDNT
6003 3390 0 RS6103 PRIV/RSDNT
```

Figure 9-8 Check source device address
3. After verifying all the source devices, check the target devices by issuing the same display command to all LPARs for the target range of devices (8100-8103). Figure 9-9 shows that all the devices are offline.

```
IEE421I RO *ALL,D U,,8100,4 216
MCECEBC RESPONSES ---------------------------------------------------------------
IEE457I 23.02.55 UNIT STATUS 215
UNIT TYPE STATUS VOLSER VOLSTATE
8100 3390 F-NRD /RSDNT
8101 3390 F-NRD /RSDNT
8102 3390 F-NRD /RSDNT
8103 3390 F-NRD /RSDNT
MZBCVS2 RESPONSES ---------------------------------------------------------------
IEE457I 23.02.55 UNIT STATUS 919
UNIT TYPE STATUS VOLSER VOLSTATE
8100 3390 F-NRD /RSDNT
8101 3390 F-NRD /RSDNT
8102 3390 F-NRD /RSDNT
8103 3390 F-NRD /RSDNT
```

**Figure 9-9  Check target device address**

4. Bring the target devices online by issuing the VARY ONLINE command to the devices as seen in Figure 9-10.

```
IEE421I RO *ALL,VARY 8100-8103,O 240
MCECEBC RESPONSES ---------------------------------------------------------------
IEE457I 23.05.42 UNIT STATUS 238
UNIT TYPE STATUS VOLSER VOLSTATE
8100 3390 O XX8100 PRIV/RSDNT
8101 3390 O XX8101 PRIV/RSDNT
8102 3390 O XX8102 PRIV/RSDNT
8103 3390 O XX8103 PRIV/RSDNT
MZBCVS2 RESPONSES ---------------------------------------------------------------
IEE457I 23.05.42 UNIT STATUS 921
UNIT TYPE STATUS VOLSER VOLSTATE
8100 3390 O XX8100 PRIV/RSDNT
8101 3390 O XX8101 PRIV/RSDNT
8102 3390 O XX8102 PRIV/RSDNT
8103 3390 O XX8103 PRIV/RSDNT
```

**Figure 9-10  Vary target devices online**
5. After all the devices (source and target) used in the data migration are online, start the TDMF TSO monitor. The member to start the monitor is in the TDMF library (SYS1.IBM.HGTD530.GTDELIB) data set. Open the data set in edit to run the REXX member as shown in Figure 9-11.

```
<table>
<thead>
<tr>
<th>Command - Enter &quot;/&quot; to select action</th>
<th>Message</th>
<th>Volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYS1.IBM.HGTD530.GTDELIB</td>
<td>VSLNSM</td>
<td></td>
</tr>
<tr>
<td>SYS1.IBM.HGTD530.GTDLIB</td>
<td>VSLNSM</td>
<td></td>
</tr>
<tr>
<td>SYS1.IBM.HGTD530.GTDMLIB</td>
<td>VSLNSM</td>
<td></td>
</tr>
<tr>
<td>SYS1.IBM.HGTD530.GTDPLIB</td>
<td>VSLNSM</td>
<td></td>
</tr>
<tr>
<td>SYS1.IBM.HGTD530.GTDTLIB</td>
<td>VSLNSM</td>
<td></td>
</tr>
<tr>
<td>SYS1.IBM.HGTD530.SAMPLIB</td>
<td>VSLNSM</td>
<td></td>
</tr>
<tr>
<td>SYS1.IBM.HGTD530.TIB</td>
<td>VSLNSM</td>
<td></td>
</tr>
</tbody>
</table>
```

**End of Data Set list**

Figure 9-11  Edit the TDMF exec file

6. Run the TDMF member to start the TDMF monitor as shown in Figure 9-12.

```
<table>
<thead>
<tr>
<th>Command - Enter &quot;/&quot; to select action</th>
<th>Message</th>
<th>Volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>TDMF</td>
<td>87</td>
<td>2011/06/30 01:33:23</td>
</tr>
<tr>
<td>TDMFMON</td>
<td></td>
<td>STRUBEL</td>
</tr>
</tbody>
</table>
**End**
```

Figure 9-12  Start the TDMF monitor
7. The TDMF startup window is displayed as seen in Figure 9-13. Press Enter to continue the process.

8. The TDMF Monitor menu is displayed as seen in Figure 9-14. There are many options presented in this menu. Each option is covered in detail in the TDMF Installation and Reference manual. Select option 0 to open SYS1.IBM.HGTD530.SAMPLIB, where you find example migration jobs.
9. The migration jobs are in the CNTL data set (specifically STXXXXX.TDMF.ITSO.TDMF.CNTL in the test environment). Edit (customize) some of the members. An example of editing the command data set that defines the new COMMDS is shown in Figure 9-15.

```
EDIT       STXXXXX.TDMF.ITSO.TDMF.CNTL(CDSL$G01) - 01.00   Columns 00001 00072
Command ===>                                                  Scroll ===> CSR
****** ************************************************************ Top of Data ************************************************************
000001 //CDSL$G01 JOB A185,'IBM',MSGCLASS=I,REGION=0M,
000002 //      CLASS=A,TIME=1440,NOTIFY=&SYSUID
000003 //   * telefon number
000004 //   * IBM name of service people- TDMF DATA MIGRATION
000005 /*JOBPARM S=CEBC
000006 //*************************************************************************
000007 // * Create SYSCOM DATA SETS AT LOCAL SITE
000008 //*************************************************************************
000009 //STEP01 EXEC PGM=IEFBR14
000010 //SYSPRINT DD SYSOUT=*  
000011 //SYSCOM DD DSN=SYS1.TDMF.ITSO.LOCAL.SYSCOM01,
000012 //      DISP=(NEW,CATLG,DELETE),
000013 //      SPACE=(CYL,70,,CONTIG),UNIT=3390,
000014 //      DCB=(LRECL=4096,BLKSIZE=4096,RECFM=F,DSORG=PS),
000015 //      VOL=SER=RS6100
000016 /*
****** ************************************************************ Bottom of Data ************************************************************
```
Chapter 9. Using TDMF for z/OS

Figure 9-16 shows examples of a MASTER and an AGENT job. For more information about parameters, see “Creating jobs” on page 426.

```plaintext
EDIT       STXXXXX.TDMF.ITSO.TDMF.CNTL(TDMLBC01) - 01.01   Columns 00001 00072
Command ====>                                                  Scroll ====> CSR
****** *********************** Top of Data ***********************
000001 //TDMLBC01 JOB A1B5,'IBM',MSGCLASS=I,REGION=OM,
000002 //      CLASS=A,TIME=1440,NOTIFY=&SYSUID
000003 /* IBM telefon number
000004 /* IBM name of service people- TDMF DATA MIGRATION
000005 /*JOBPARM S=CEBC
000006 //******************************************************************************
000007 // Master JOB TDMF running: LOCAL
000008 //******************************************************************************
000009 //LOCAL01  EXEC PGM=GTDMAIN,PARM=MASTER,REGION=OM
000010 //STPLIB  DD DISP=SHR,DSN=SYS1.IBM.HGTD530.GTDLLIB
000011 //GTDKEY   DD DISP=SHR,DSN=SYS1.IBM.HGTD530.GTDLLIB
000012 //SYSCOM   DD DISP=SHR,DSN=SYS1.TDMF.ITSO.LOCAL.SYSCOM01
000013 //SYSPRINT DD SYSOUT=*  
000014 //SYSNAP  DD SYSOUT=*  
000015 //SYSSIN DD *
000016 SESSION M(CEBC) AGENT(CVS2)
000017     OPT(FAST UNIDENT(T) NOPROMPT CONC(2)
000018         PAC CHECKT)
000019 MIGRATE RS6000 XX8100 YY6000
000020 MIGRATE RS6001 XX8101 YY6001
000021 MIGRATE RS6002 XX8102 YY6002
000022 MIGRATE RS6003 XX8103 YY6003
****** **************************** Bottom of Data *****************************

****** *********************** Top of Data ***********************
000001 //TDALCV01 JOB A1B5,'IBM',MSGCLASS=I,REGION=OM,
000002 //      CLASS=A,TIME=1440,NOTIFY=&SYSUID
000003 /* IBM telefon number
000004 /* IBM name of service people- TDMF DATA MIGRATION
000005 /*JOBPARM S=CVS2
000006 //******************************************************************************
000007 // AGENT JOB TDMF running: CVS2
000008 //******************************************************************************
000009 //STEP01 EXEC PGM=GTDMAIN,PARM=AGENT,REGION=OM
000010 //STPLIB  DD DISP=SHR,DSN=SYS1.IBM.HGTD530.GTDLLIB
000011 //GTDKEY   DD DISP=SHR,DSN=SYS1.IBM.HGTD530.GTDLLIB
000012 //SYSCOM   DD DISP=SHR,DSN=SYS1.TDMF.ITSO.LOCAL.SYSCOM01
000013 //SYSPRINT DD SYSOUT=*  
000014 //SYSSDUMP DD SYSOUT=*  
000015 //SYSSNAP  DD SYSOUT=*  
****** **************************** Bottom of Data *****************************

******************************************************************************
** Start AGENT, on for each LPAR connected to source and target
******************************************************************************
```

Figure 9-16  Example: submit Master and Agent jobs
10. Before starting, run the job to create the COMMDS. Figure 9-17 shows a new query of the data sets that displays the COMMDS (SYS1.TDMF.ITS0.LOCAL.SYSCOM01).

Figure 9-17 Checking for new COMMDS

DSLIST - Data Sets Matching SYS1.TDMF.ITS0

Command ==>                         Scroll ==> CSR

Command - Enter "/" to select action  Message       Volume
-------------------------------------------------------------------
SYS1.TDMF.ITS0.LOCAL.SYSCOM01        RS6100
**************************************** End of Data Set list ****************************************

11. After verifying that the new COMMDS exists, submit the master and agent jobs as shown in Figure 9-16 on page 413.

12. Change to the TDMF TSO monitor using option 1 in the Monitor menu as seen in Figure 9-14 on page 411. Option 1 allows you to monitor the progress of the current session (Figure 9-18). In this example, the data is being migrated on four volumes. Two of the volumes are copying at the same time. The progress monitor is updated to show the migration phases and the percentage of completion for each phase after pressing the Enter key. You also get an estimated end time for the initial copy phase for each volume.

Figure 9-18 Monitor the session
Figure 9-19 displays the output of the session monitor with the first two volumes in a Refresh phase at 100% complete. The copy has not yet started for the next two volumes.

![Session Monitor](image1)

**Figure 9-19  Refresh phase started**

Figure 9-20 shows that the first two volumes are completed, and the next two volumes are in the copy phase. Based on the parameter (CONCurant=2), only two volumes run at a time. However, this parameter can be dynamically changed in the monitor.

![Session Monitor](image2)

**Figure 9-20  Complete and copy phases**
Figure 9-21 illustrates the Refresh to Synchronize phase for the second set of volumes.

<table>
<thead>
<tr>
<th>Session Monitor</th>
<th>Row 1 to 8 of 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command ===&gt;</td>
<td>Scroll ===&gt; PAGE</td>
</tr>
</tbody>
</table>

Softek TDMF Master V5.3.0 Session Active.
ComDataSet : SYS1.TDMF.ITSO.LOCAL.SYSCOM01

<table>
<thead>
<tr>
<th>Source</th>
<th>Migration</th>
<th>Percent Complete</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group</td>
<td>VolSer</td>
<td>Phase</td>
</tr>
<tr>
<td>RS6000</td>
<td>Complete</td>
<td></td>
</tr>
<tr>
<td>RS6001</td>
<td>Complete</td>
<td></td>
</tr>
<tr>
<td>RS6002</td>
<td>Refresh 1</td>
<td>-----------------</td>
</tr>
<tr>
<td>RS6003</td>
<td>Refresh 1</td>
<td>-----------------</td>
</tr>
</tbody>
</table>

Softek TDMF Master V5.3.0 Session Active.
ComDataSet : SYS1.TDMF.ITSO.LOCAL.SYSCOM01

<table>
<thead>
<tr>
<th>Source</th>
<th>Migration</th>
<th>Percent Complete</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group</td>
<td>VolSer</td>
<td>Phase</td>
</tr>
<tr>
<td>RS6000</td>
<td>Complete</td>
<td></td>
</tr>
<tr>
<td>RS6001</td>
<td>Complete</td>
<td></td>
</tr>
<tr>
<td>RS6002</td>
<td>Quiescing</td>
<td>-----------------</td>
</tr>
<tr>
<td>RS6003</td>
<td>Quiescing</td>
<td>-----------------</td>
</tr>
</tbody>
</table>

Softek TDMF Master V5.3.0 Session Active.
ComDataSet : SYS1.TDMF.ITSO.LOCAL.SYSCOM01

<table>
<thead>
<tr>
<th>Source</th>
<th>Migration</th>
<th>Percent Complete</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group</td>
<td>VolSer</td>
<td>Phase</td>
</tr>
<tr>
<td>RS6000</td>
<td>Complete</td>
<td></td>
</tr>
<tr>
<td>RS6001</td>
<td>Complete</td>
<td></td>
</tr>
<tr>
<td>RS6002</td>
<td>Synchronize</td>
<td>-----------------</td>
</tr>
<tr>
<td>RS6003</td>
<td>Synchronize</td>
<td>-----------------</td>
</tr>
</tbody>
</table>

Figure 9-22 shows that all the copies for this session are complete. Remember that this migration was a simple one involving only the four volumes. A typical migration scenario usually involves many more volumes.

<table>
<thead>
<tr>
<th>Session Monitor</th>
<th>** no sessions active **</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command ===&gt;</td>
<td>Scroll ===&gt; PAGE</td>
</tr>
</tbody>
</table>

No Softek TDMF sessions active.

Figure 9-22  TDMF session complete
13. Check the state of the volumes using the display command again. Figure 9-23 lists the output in the syslog (only the user syslog in these examples). As you can see, the old source volumes 6100-6103 are now offline to the LPARs. The volume ID is displayed even if the device is offline, which is expected at the completion of the TDMF data migration.

Figure 9-23 Query old source volumes after migration

Figure 9-24 shows the output of a query against the old target volumes (8100-8103). These volumes are now the source volumes and online to the LPARs. The data migration is complete.

Figure 9-24 Verify that the new source volumes are online
9.3 TDMF preferred practices

This section offers general advice and preferred practices for TDMF z/OS.

9.3.1 Keeping current

Make sure before any migration that your TDMF z/OS code is up-to-date. Use the following techniques to keep your code current:

- Bookmark the TDMF Technical Support website. It contains information that can help you successfully run TDMF.
- Register for automatic email notification. Registration ensures that you are notified whenever the Technical Support website is updated. You receive the actual link for registration with your license key (by mail).
- Click Notification Registry on the Technical Support website to get an IBM Lotus® Notes® Group ID.
- Download each program temporary fix (PTF) as it becomes available. If you miss one, however, the list fixes are still available. All PTFs are available for a SMP/E installation only. NON SMP/E installation requires a new download of the code and reinstallation. See the Installation and Reference Manual at:
- Manuals are periodically updated. Make sure that you have the most current information available by checking for Technical Information Bulletins (TIBs). TIBs are typically issued between manual releases. They are available at:

9.3.2 Setting default options

As part of the post-installation tailoring of TDMF, option defaults are set by the SYSOPTN batch job and the loading of the software keys. If you want to use pacing in every migration session, set this value to YES in the SYSTOPTN batch job during first activation. That value is the default for all subsequent sessions.

Values set in the SYSOPTN batch job can be overridden within a session. The order of overriding of options is as follows:

- Session statement
- Group statements
- Migrate statements

In other words, values in the session statement override the installation defaults. Values in the group statement override the session and installation defaults. And finally, the migrate statements override all of the above.

For more information, see the TDMF Installation Manual, Chapter 2 “TDMF Control Statements” at:
9.3.3 Storage requirements

In this example, there are 16 3390-3 volumes being migrated with two systems involved, and Offline Volume Access is not selected. In this case, the storage requirements for the master and agent systems are:

- **Master system:**
  - Fixed common storage: 612K ECSA, 9K ESQA, 256-bytes CSA
  - Pageable common storage: 28k ECSA
  - Fixed extended private area storage: 16,600K

- **Agent system:**
  - Fixed common storage: 617K ECSA, 5K ESQA, 256-bytes CSA
  - Pageable common storage: 28k ECSA
  - Fixed extended private area storage: 656K

If the Compare option or Full Speed Copy is requested, an additional 900K buffer for each volume migration is allocated in ECSA. For more information about storage requirements, see the *TDMF Installation Manual*, Chapter 2 “Storage Requirements” at:


9.3.4 Communications data set

Each TDFM session requires a unique communications data set (COMMDS). The master and agent systems communicate to each other using the COMMDS. The COMMDS cannot be on the same volume involved in the session. Allocate disk space for COMMDS on a separate, NONSMS-managed volume. The COMMDS must be placed on a device that supports CKD/E.

**Important:** If another session is submitted using the same COMMDS name while the first session is running, unpredictable results occur.

If a problem occurs, the COMMDS is the primary tool for problem determination and resolution. Although reuse of a COMMDS is normal, be aware that the previous session data is no longer available. If there is a problem, do not reuse the COMMDS so it is available for problem determination and resolution.

**Tip:** Use a new COMMDS for each session, so that if a problem occurs, all information is available for analysis.

Member ALLOCCM in SAMPLIB allocates the COMMDS. This data set must be physically located on a cylinder boundary with contiguous space. For an example, see Figure 9-15 on page 412. Volumes containing COMMDS cannot be moved by TDFM.

**COMMDS reserve**

TDFM periodically issues a RESERVE macro for the COMMDS to serialize communication between the master and agent systems. For details, see the *TDMF Installation Manual*, Chapter 4 “Unicenter CA-MIM Resource Sharing or Global Resource Serialization.”

**Sizing a COMMDS**

To calculate the size of a COMMDS, use the following formula:

\[ \text{COMMDS CYLS} = V \times (S + K) \]
Where \( V \) is the number associated with the number of volumes as follows:
- 64 volumes = 2.5
- 128 volumes = 5.0
- 256 volumes = 7.5
- 512 volumes = 10.0

In addition, \( S \) is the number of participating systems, and \( K \) is the size of the source volumes involved as follows:
- For 3390-3, \( K = 4 \)
- For 3390-9, \( K = 6 \)
- For 3390-27, \( K = 15 \)

For example, consider a system which contains 128 3390-3 and 128 3390-9 volumes across 8 LPARs. Using the largest device type in the session (and therefore \( K = 6 \)), the size is calculated as follows:
\[
\text{CYLS} = 7.5 \times (8 + 6) \quad \text{(always use the largest device type in session)}
\]
\[
\text{CYLS} = 7.5 \times 14
\]
\[
\text{CYLS} = 105 \quad \text{(round down if required)}
\]

**Data set type and location for COMMDS**
The COMMDS can be defined as a generation data group (GDG), which can ease the tracking of these data sets. In this case, each new generation must be created before running the session. Using GDG allows reference within the JCL to relative generation zero (0) and the base name to be specified on the SESSION control statement.

**Tip:** GDG data sets are not preferred because they can be overwritten during the migration process. Instead, use sequential data sets, which are the default.

For more details, see the *TDMF Installation Manual*, Chapter 3 “Placement of the Communications data set.”

### 9.3.5 Participation of agent systems

In a shared DASD environment, all Logical Partitions (LPARs) must have access to the source and target devices participating in the TDMF session. Each master or agent system is only aware of I/O to the source or target device from its perspective. If a session is run without inclusion of all LPARs connected to the source or target devices, data integrity is compromised.

Since TDMF version 5.2.0, you can run an AGENT while the source and target device are offline to the LPAR. Therefore, you do not have to split the volumes into groups where the volumes have the same status to the LPARs. Some volumes (source and target) can be offline in one LPAR and the session will not terminate. For more information, see the *TDMF Installation Manual*.

The Unidentified Systems option can be used to verify that the correct number of agent systems are being run. The default action is set when the SYSOPTN job is run, but can be overridden using the options within the SESSION control statement. The options available are:
- **Terminate on Error:** This issues an error message and terminates the migration/replication (RC08).
- **Warning:** This issues a warning message, but the migration/replication continues (RC04).
### 9.3.6 Protection of target volume data

If a target volume contains data, TDMF overwrites that data as long as no files are open at the time the session starts. If a file is open on a target volume at volume initiation, TDMF terminates that migration session. The system displays return code 12 (RC 12) and any error messages pertinent to the situation.

The CHECK TARGET option ensures that TDMF does not overwrite data on a target device. Selection of this option informs TDMF that only the VTOC, VTOCIX, and VVDS entries are allowed on the target volume.

**Tip:** Use the CHECK TARGET option as default to protect your data. Set the CHECK TARGET option in each session control statement.

For details, see the *TDMF Installation Manual*, Chapter 2 “Session Options.”

### 9.3.7 Pacing

Always use pacing so that application performance is not affected. Some volumes regularly experience high I/O rates.

The following options are available to regulate pacing:

- Reverse pacing
- Full speed copy
- FastCopy
- Active in Copy

Pacing is active during the Copy and Refresh phases only. After the volumes or session enter the Quiesce phase and continue through the Termination phase, pacing is not started. For more details, see the *TDMF Installation Manual*, Chapter 1 “Major Phases of Migration.”

**Tip:** Generally, use the default pacing options. Move frequently updated volumes during a low activity time period.

**Reverse pacing**

Reverse pacing starts at one track per I/O operation, and scales upwards depending on the activity on the source volume. For more details, see the *TDMF Installation Manual*, Chapter 3 “Dynamic Volume Pacing.”

---

Informational: This issues an informational message and the migration/replication continues (RC00).

This parameter works with only 3990-6 control units or later (2105/2107). If you are using 3990-3 (old HW), TDMF cannot check the attached systems.

**Important:** You are responsible for running the correct number of AGENTs.
Full speed copy
This option causes TDMF to increase the number of buffers from one to two to allow interleaving of the I/O. Using two buffers shortens the migration time. Use this option judiciously because the additional load to the source and target volumes can negatively affect applications and the migration itself. Field statistics show that the decrease in migration time is 40% on average. For more details, see the *TDMF Installation Manual*, Chapter 2 “Full Speed Copy Impact.”

FastCopy
FastCopy instructs TDMF to copy only the allocated tracks and cylinders, ignoring empty ones. This option can decrease the amount of time required to migrate a volume with no performance impact. Use the FastCopy option at all times. For more details, see the *TDMF Installation Manual*, Chapter 2 “Common Options Table.”

Active in Copy
The volume pairs in a migration are considered active from volume initialization to volume termination. Active in Copy instructs TDMF to treat a volume as being active from volume initialization through the completion of the first refresh pass. Use of this option limits the number of active copy tasks but speeds the normal copy process. For more details, see the *TDMF Installation Manual*, Chapter 2 “Session Options.”

9.3.8 Rank contention and storage subsystem performance
When large numbers of volumes are to be copied, it is important to understand the following about the system:
- The architecture of the subsystem
- The logical to physical mapping of the volumes within the array

Use this information during the planning, coding, and execution of TDMF sessions. This information helps ensure that these sessions do not adversely affect application, array, and subsystem performance.

When experimenting, do not migrate more than two volumes from a logical control unit (LCU). Using monitors such as RMFMON, check if there is an impact to the system performance. Also, check channel utilization to make sure that it is no more than 70-75% per channel path. Change the number of concurrent running volumes based on the results using the TDMF monitor.

For more details, see the *TDMF Installation Manual*, Chapter 3 “Raid Subsystems and Rank Contention.”

9.3.9 TDMF interaction with other programs
Before the use or execution of TDMF, see the *TDMF Installation Manual*, Chapter 4: “Planning Considerations.” Ensure that the conditions are in place for the coding and execution of the TDMF sessions.

If you use an array-based mirroring solution in your environment, it must be protected. Protection is especially important if you need to keep that environment active during a migration. In this case, review the following topics in Chapter 4:
- Migrating from SRDF to PPRC Volumes
- Migrating to SRDF Volumes
- Peer-to-Peer Remote Copy
9.3.10 Identification of volumes requiring special handling

Some volumes might require special handling before, during, or after a migration. These volumes need to be identified and potentially put into one or more separate sessions.

In addition, some volume migrations require an application recycle to complete the migration. For more information, see the TDMF Installation Manual, Chapter 4: “Planning Considerations.”

Important: Do not move active Local Page data sets or active Sysplex Couple data sets.

Migrating Local Page data sets

Local Page data sets cannot be moved while active. To migrate them, perform the following steps:

1. Analyze location of these data sets using the D ASM command at each LPAR accessing the source/target storage (Example 9-1).

   Example 9-1 Display Local Page data sets
   
   COMMAND INPUT ===> /D ASM
   RESPONSE=MCECEBC
   IEE200I 11.32.10 DISPLAY ASM 536
   TYPE    FULL STAT   DEV DATASET NAME
   PLPA     82%   OK  5249 PAGE.MCECEBC.PLPA
   COMMON   0%    OK  5249 PAGE.MCECEBC.COMMON
   LOCAL    0%    OK  5249 PAGE.MCECEBC.LOCAL1
   LOCAL    0%    OK  524A PAGE.MCECEBC.LOCAL2
   LOCAL    0%    OK  5249 PAGE.MCECEBC.LOCAL3
   LOCAL    0%    OK  524A PAGE.MCECEBC.LOCAL4
   PAGEDEL COMMAND IS NOT ACTIVE

2. Check whether the Local Page data sets are on separate volumes. If not, create temporary Local Page data sets on different volumes (old storage subsystem).

3. Release all active Local Page data sets on the same volume using the PAGE DELETE command. It takes time to release them from the system. Use the DISPLAY ASM command to check on the progress.

4. When all Local Page data sets on the same volume are released from the system Page Pool, migrate that volume.

5. After migrating the volume, use the PAGE ADD command to add Local Page data sets to the System Page Pool again.

6. Repeat the proceeding steps for all volumes where Local Page data sets are located.

PLPA and COMMON Page data sets can be moved by TDMF, but must be moved separately.

Migrating JES spool and check point data sets

JES2 spool and checkpoint data sets cannot be moved with normal data. Mark all volumes containing these data sets in your volume migration list and migrate them in the same session. Run one volume at the time, and at a time when there is not a heavy I/O load on the
system. For more details on systems running JES3, see the *TDMF Installation and Reference Manual*.

### Migrating Couple data sets (sequence)

Analyze the volume where the Couple data sets are located using the command `D XCF, COUPLE, TYPE=SYSPLEX` as shown in Example 9-2.

**Example 9-2 Display sysplex Couple data sets**

<table>
<thead>
<tr>
<th>COMMAND INPUT</th>
<th>SCROLL</th>
</tr>
</thead>
<tbody>
<tr>
<td>==&gt; /D XCF,COUPLE,TYPE=SYSPLEX</td>
<td>CSR</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RESPONSE=MCECEBC</th>
</tr>
</thead>
<tbody>
<tr>
<td>IXC358I 22.08.02 DISPLAY XCF 886</td>
</tr>
</tbody>
</table>

**SYSPLEX COUPLE DATA SETS**

<table>
<thead>
<tr>
<th>PRIMARY</th>
<th>DSN: SYS1.CEBCPLEX.XCF.CDS01</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOLSER: COUPL1</td>
<td>DEVN: B23D</td>
</tr>
<tr>
<td>FORMAT TOD MAXSYSTEM MAXGROUP(PEAK) MAXMEMBER(PEAK)</td>
<td></td>
</tr>
<tr>
<td>12/09/2010 10:43:02</td>
<td>32 200 (39) 303 (8)</td>
</tr>
</tbody>
</table>

**ADDITIONAL INFORMATION:**
- All types of Couple data sets are supported
- GRS STAR mode is supported
- Cluster resource management is supported
- System status detection protocol is supported

<table>
<thead>
<tr>
<th>ALTERNATE</th>
<th>DSN: SYS1.CEBCPLEX.XCF.CDS02</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOLSER: COUPL2</td>
<td>DEVN: B23E</td>
</tr>
<tr>
<td>FORMAT TOD MAXSYSTEM MAXGROUP MAXMEMBER</td>
<td></td>
</tr>
<tr>
<td>12/09/2010 10:46:44</td>
<td>32 200 303</td>
</tr>
</tbody>
</table>

**ADDITIONAL INFORMATION:**
- All types of Couple data sets are supported
- GRS STAR mode is supported
- Cluster resource management is supported
- System status detection protocol is supported

Check whether these data sets are on separate volumes. If not, do not move these volumes because the time-out interval of the timer accessing the Couple Data Set (CDS) default is 15 seconds.

To check volumes also for Local Page data sets, perform the following steps:

1. Switch the couple data set from primary to alternate:
   ```
   setxcf couple,type=sysplex,pswitch
   ```

2. Migrate the volume (COUPL1) with the non-active old primary couple data set.

3. Add the old primary to the system as the new alternate couple data set:
   ```
   setxcf couple,type=sysplex,acouple=SYS1.CEBCPLEX.XCF.CDS01
   ```

4. Switch the couple data set again:
   ```
   setxcf couple,type=sysplex,pswitch
   ```

5. Migrate the volume (COUPL2) with the non-active old alternate couple data set.

6. Add the old alternate to the system as the new alternate again:
   ```
   setxcf couple,type=sysplex,acouple=SYS1.CEBCPLEX.XCF.CDS02
   ```
9.3.11 Migration considerations

Consider the following tips when planning your migration:

- Consider using a naming convention for the batch jobs. If a number of sessions are to be run, use a naming convention that groups the master, agents, and COMMDS in an easy-to-find format. For more information, see Chapter 11, “Using TDMF TCP/IP for z/OS” on page 483.

- The keyword TIME=1440 or TIME=(mm,ss) can be specified on job cards to avoid system ABEND 322 (S322). TIME=1439 is preferable because TIME=1440 disables SMF time recording for that job. TIME=(mm,ss) allows recording of SMF times but limits the amount of processor time used by the TDMF session. Ensure that a reasonable amount of time is allowed for each TDMF session, especially if the COMPARE option is used. See members MASTER and AGENT in the SAMPLIB for examples. For more information about the TIME parameter, see the MVS JCL manuals in the z/OS Internet Library at:

- Always code TDMF sessions for as many pairings as possible. The real storage requirements, based on the number of concurrent volumes specified, remain the same regardless of the number of volumes within the session. By coding the maximum number of volumes per session, the management of multiple sessions is lessened.

- Set Active in Copy and the Number of Concurrent Volumes to the number of channels the LPAR can access from the source subsystem. In a typical technology refresh, the target subsystem has newer storage devices and newer channels, such as FICON versus ESCON connections. Because ESCON channels are slower than FICON channels, limit the number of active volumes based on the slower channel paths. A general rule of thumb for ESCON channel paths is two active volumes per path.

- Employ FastCopy whenever possible. This option copies only the allocated tracks and cylinders of the device: No empty tracks are copied.

- Use Full Speed Copy if you want to employ double buffering. When this option is selected, two 900 K buffers are allocated and pacing is automatically started. Overall time reduction is dependent on a number of factors, but an average of 45 percent can be expected.

- Select the Dynamic ICKDSF REFVTOC option. ICKDSF is dynamically called at the end of the migration if invocation is necessary to reformat the VTOC. If no changes are required, the call is not issued.

- Select Toleration of Invalid Count Fields to ensure that volumes with invalid count fields (CCHH mismatch) are copied without error. These count fields are not corrected.
Creating jobs

Figure 9-25 shows creating a COMMDS.

```plaintext
//TDMLBC04 JOB (A185,SY), 'STXXXX', 
  NOTIFY=&SYSUID,REGION=OM, 
  CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
/*JOBPARM SYSAFF=CEBC

//define new communication dataset
// please set correct size in CYL
// depending on number of volumes to move.
// See TDMF Installation manual for details.

//DEFCOX04 EXEC PGM=IEFBR14
//SYSPRINT DD SYSOUT=* 
//DASDI1 DD DSN=SYS1.TDMF.ITSO.LOCAL.SYSCOM04,DISP=(NEW,CATLG), 
  VOL=SER=RS6100,STORCLAS=NONSMS, 
  UNIT=3390,SPACE=(CYL,70,,CONTIG) 
/*
```

Figure 9-25  JCL example to create COMMDS

Sample job naming conventions are shown in Example 9-3. In a migration service, be aware that you can run several jobs in parallel. A good idea is to reflect the type, LPAR name, and the session number in the job name.

Example 9-3  Job Naming conventions

```plaintext
//TDMLBC04 JOB ..... 
!!! !
!!! --> session number (2 digits)
!!!+----> 2 char out of the LPAR name (first or last 2 char)
!!    to identify AGENTs at different LPARs
!+----> L=local
+------> M=Master, A=AGENT
```
Figure 9-26 shows creating a MASTER task.

```plaintext
//TDMLBC04 JOB (A204,SYS),'STXXXX',
// NOTIFYSYSUID,REGION=OM,
// CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
/*JOBPARM SYSAFF=CEBC
 ** define master job
 **%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 //MIGRAT04 EXEC PGM=GTDMAIN,PARM=MASTER,TIME=1440,REGION=OM
 //STEPLIB DD DISP=SHR,DSN=SYS1.IBM.HGTD530.GTDLLIB
 //SECCOM DD DISP=SHR,DSN=SYS1.IBM.HGTD530.GTDLLIB
 //SYSCOM DD DISP=SHR,DSN=SYS1.TDMF.ITSO.LOCAL.SYSCOM04
 //SYSPRINT DD SYSOUT=*  
 //DSFPRINT DD SYSOUT=*  
 //SYSUDUMP DD SYSOUT=* 
 //SYSSNAP DD SYSOUT=* 
 //SYSSIN DD *  
 SESSION
 MASTER(CEBC)
 AGENT(CVS2)
 OPTIONS(CONCURRENT(02))
 UNIDENTIFEDS(TERMINATE)
 CHECKTARGET
 FASTCOPY
 PACING(NORMAL)
 IKDSF
 NOCONFIRM
 NOPROMPT
 NOALLOWINVALIDCOUNTS
 NOCOMPARE
 )
 MIGRATE RS6000 XX8100 YY6000
 MIGRATE RS6001 XX8101 YY6001
 MIGRATE RS6002 XX8102 YY6002
 MIGRATE RS6003 XX8103 YY6003
 */
```

Figure 9-26  JCL example for master system
Figure 9-27 shows creating an AGENT task. The master and agent tasks together are called a SESSION.

```
//TDALS204 JOB (A185,SYS), 'STXXXX',
//     NOTIFY=&SYSUID,REGION=OM,
//     CLASS=A, MSGCLASS=X, MSGLEVEL=(1,1)
/*/JOBPARM SYSAFF=CVS2
/****************************************************
//** Start AGENT, one for each LPAR connected to source and target
/****************************************************
//AGENT04 EXEC PGM=GTDMAIN, PARM=AGENT
//STELIB DD DSN=SYS1.IBM.HGTD530.GTDLLIB, DISP=SHR
//SECCOM DD DSN=SYS1.IBM.HGTD530.GTDLLIB, DISP=SHR
//SYSCOM DD DSN=SYS1.TDMF.ITSO.LOCAL.SYSCOM04, DISP=SHR
//SYSPRINT DD SYSOUT=* 
//SYSSDUMP DD SYSOUT=* 
//SYSSNAP DD SYSOUT=* 
//SYSIN DD DUMMY
/*
```

**Important:** Do not forget that AGENTs must run on all LPARs. Also, make sure that the volumes are accessed and online.

### 9.3.12 Estimating how long it takes to move the data

How long it takes to move the data depends on a number of factors. Some of the factors are the options selected for a specific volume migration or for the entire session. Others are the activity of the processor, channel paths, and devices. This section addresses the following factors:

- **Pacing**
- **Number of sessions**
- **Factors external to TDMF**
- **Channel paths**
- **Storage subsystem cache**
- **Device size and rotational speed**
- **Logical to physical mapping**
- **In summary**

#### Pacing

Standard I/O pacing is 15 tracks per I/O operation. TDMF samples device activity every 30 seconds during the copy and refresh phases. If device or channel path activity affects the production environment, TDMF dynamically adjusts the number of tracks read/written in a single I/O operation. You can also use reverse pacing, which is designed to be used when moving a volume with high activity. This option starts at one track per I/O operation and, if activity allows, increase the number of tracks per I/O dynamically.

User-specified pacing allows you to determine the number of tracks read/written in a single I/O operation. The values allowed are five, three, and one track per I/O. This value might or might not be static depending on whether pacing is specified.
If an operating system is heavily used, the number of available real storage frames can be limited. TDMF tracks this value so that it does not affect production from that standpoint. If real storage pacing is started, I/O pacing is automatically affected.

To speed up the migration, you can select Full Speed Copy. This option allows the system to use two buffers, increasing speed by around 45% on normal systems. See the Migration Recommendations section of the TDMF Installation Manual for more detail.

You can also adjust the number of concurrent volumes. The current release of TDMF allows up to 512 volumes to be coded per session. However, the number of volumes multiplied by number of LPARs must not exceed 2048. Set this with the Number of concurrent volumes option. Another option that can be used with the number of concurrent volumes is Active-in-Copy. This option saves time in the overall session.

The number of concurrent running volumes can be changed dynamically using TDMF monitor item 2 as shown in Figure 9-28. Select the field and change it, then press Enter to activate the change. If you reduce the number, the running volumes continue copying until the end of the volume. New volumes start when the new defined value of volumes is reached.

![Figure 9-28 Changing the number of concurrent volumes dynamically](image)
You can manage a single volume from this TDMF monitor panel. Select the volume and use one of the listed line commands as shown in Figure 9-29.

<table>
<thead>
<tr>
<th>Transparent Data Migration Facility</th>
</tr>
</thead>
<tbody>
<tr>
<td>PF3=End</td>
</tr>
</tbody>
</table>

Following valid on all Systems

<table>
<thead>
<tr>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display</td>
</tr>
<tr>
<td>Migration Messages             : M</td>
</tr>
<tr>
<td>Display</td>
</tr>
<tr>
<td>Migration Performance Data     : D</td>
</tr>
<tr>
<td>Display</td>
</tr>
<tr>
<td>Volume's Offline Volume Access (OVA) Job(s) : X</td>
</tr>
</tbody>
</table>

Following valid only on MASTER System

<table>
<thead>
<tr>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acknowledge</td>
</tr>
<tr>
<td>Mirror Change on Volume Swap    : A</td>
</tr>
<tr>
<td>Approval</td>
</tr>
<tr>
<td>Approve this migration           : B</td>
</tr>
<tr>
<td>Continue</td>
</tr>
<tr>
<td>Volume Migration or its Group    : C or CG</td>
</tr>
<tr>
<td>Confirm</td>
</tr>
<tr>
<td>PPIT Recycle for Volume or its Group : R or RG</td>
</tr>
<tr>
<td>Confirm</td>
</tr>
<tr>
<td>Start Volume Migration or its Group : F or FG</td>
</tr>
<tr>
<td>Confirm</td>
</tr>
<tr>
<td>Synchronization for Volume or its Group : P or PG</td>
</tr>
<tr>
<td>Disallow</td>
</tr>
<tr>
<td>OVA registrations for Volume or its Group : H or HG</td>
</tr>
<tr>
<td>Discontinue</td>
</tr>
<tr>
<td>PPIT migrations at &quot;Recycle&quot; prompt : E or EG</td>
</tr>
<tr>
<td>Reinitialize</td>
</tr>
<tr>
<td>Migration of terminated Volume   : I</td>
</tr>
<tr>
<td>Set Goal</td>
</tr>
<tr>
<td>Synchronization Goal for Volume or its Group : Z or ZG</td>
</tr>
<tr>
<td>Suspend</td>
</tr>
<tr>
<td>Volume Migration or its Group    : S or SG</td>
</tr>
<tr>
<td>Terminate</td>
</tr>
<tr>
<td>Volume Migration or its Group    : T or TG</td>
</tr>
<tr>
<td>Terminate not allowed at APPROVAL PROMPT</td>
</tr>
</tbody>
</table>

PF4            Display the next active session

PF6            Toggle key for displaying the session options

PF10           Immediately display the session progress monitor

PF11           Toggle key for filtering volumes

**Figure 9-29   TDMF line commands**

**Number of sessions**
There is no limit to the number of TDMF sessions that can be started. The determining factor is the number of available resources each LPAR has that is involved in the migration.

**Factors external to TDMF**
When TDMF is moving data, it uses processor resources and peripheral resources such as channel paths and storage subsystem resources. These resources also determine how fast data is moved.

**Channel paths**
TDMF interacts with the Input/Output Supervisor, which manages transport mechanism or channel paths. Various channel paths that are in use today, from older technologies such as parallel paths to newer ones such as FICON. Each type of channel has a specific limit as to the amount of data that can be transported.
Table 9-1 provides raw and approximate throughput values for the types of channel paths. On most systems, FICON is not the bottleneck.

Table 9-1 IBM S/390® channel throughput for data migration

<table>
<thead>
<tr>
<th>Channel Type</th>
<th>Raw throughput</th>
<th>Approximate throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parallel (gray) old</td>
<td>3.0 MBps</td>
<td>2.3 MBps</td>
</tr>
<tr>
<td>Parallel (blue) old</td>
<td>4.5 MBps</td>
<td>3.8 MBps</td>
</tr>
<tr>
<td>ESCON</td>
<td>20 MBps</td>
<td>12 MBps</td>
</tr>
<tr>
<td>FICON</td>
<td>100 MBps</td>
<td>85 MBps</td>
</tr>
<tr>
<td>FICON</td>
<td>200 MBps</td>
<td>170 MBps</td>
</tr>
<tr>
<td>FICON</td>
<td>400 MBps</td>
<td>330 MBps</td>
</tr>
</tbody>
</table>

Storage subsystem cache
The size of cache in a storage subsystem has increased over the years. However, when a migration or migration service is being performed, it frequently involves a technology refresh. Older technology has smaller cache sizes compared to that of new technology. Although TDMF issues the channel command Inhibit Cache Load during the read operation, some storage subsystems do not honor this command. Therefore, while cache is being used by the production environment, there is the additional load of the data being moved by TDMF.

Device size and rotational speed
Over the years the size of devices has increased along with the rotational speed of the devices. However, you often migrate off older technology, which is generally slower. Know the source subsystem and all the intricacies that go with it.

Logical to physical mapping
In an RAID-1 environment, there are multiple logical devices that are mapped to a single physical device. An RAID-5 environment means that there are multiple logical devices mapped across multiple physical devices. Although these new devices are much faster than older technology, there is still only one head-disk assembly per physical device. Moving too many logical devices at once can be just as devastating to a production environment as moving data without specifying pacing. However, if Solid State Disks (SSD) are installed, the internal interfaces can be the bottleneck instead of the disks themselves.

In summary
The best migration practice is to plan. Every migration initiative, especially large migrations, should be 90% planning and 10% implementation. Understanding the production environment and when the heaviest loads occur is a part of panning, along with the impact of old and new technology. Remember, data can be moved only as fast as the slowest device or slowest channel allows.

9.3.13 Terminating a TDMF session
With the TDMF TSO Monitor or Batch Monitor, you can terminate a specific volume pairing, volume groups, or all volumes in a session dynamically. A termination request for a volume pairing might take up to a minute to be processed by TDMF. If the master and all the agent systems are active in the same IBM Parallel Sysplex®, the interval is less.
Under extreme conditions, you can use the MVS CANCEL command. If the master system fails leaving an agent system active, allow a 15-minute interval to allow the agent system to shut down automatically. If it does not, cancel the agent system job manually.

If a TDMF session still hangs after MVS Cancel, use `Display GRS,C` to determine whether canceling another job might break the deadlock. When a TDMF job is canceled, the program signals the other systems in the session to terminate the active volumes and clean up resources. Canceling for a second time, or using the MVS Force command, bypasses this recovery and causes the TDMF RTM Resource Manager routine to receive control. Canceling this way might result in migration volumes being varied offline and boxed as part of cleanup. If you issue a Force or second Cancel command, run the TDMFCLUP program as described in the chapter “Batch Utilities” of the *TDMF Installation Manual*.

**Important:** During an error, avoid issuing a CANCEL command using SDSF against TDMF master or agents. Instead, use the TDMF monitor option 2 (interaction with TDMF) to terminate a migration session.
z/OS Dataset Mobility Facility (zDMF) is a host-based (mainframe) data migration facility that migrates data at the logical data set (extent) level between sets of volumes. All data movement is accomplished without application downtime.

This chapter outlines the steps needed to conduct a successful data migration project using zDMF. It also describes the standards and provides guidelines when using zDMF for a data migration project.

zDMF is a data set level migration tool that updates the information in the ICF catalog and interacts with SMS. Therefore, there are sections that address its role in the migration process, detailed explanations of the environmental requirements, and reference materials.

This chapter contains the following sections:

- z/OS Dataset Mobility Facility overview
- zDMF migration process
- Important zDMF-related topics
- Example migration in a shared DB2 test environment
10.1 z/OS Dataset Mobility Facility overview

The task of combining volumes into new, large volume capacity storage subsystems is difficult and time-consuming. Historically combining volumes required taking revenue-generating and often mission-critical applications offline for the duration of the migration. z/OS Dataset Mobility Facility (zDMF) allows you to migrate and consolidate data sets while minimizing or eliminating application downtime.

zDMF moves data sets without disrupting access to applications during the migration. zDMF also provides the following advantages:

- Makes it easy to combine smaller capacity volumes to better use storage
- Helps to increase available system resources
- Supports implementation of tiered storage
- Improves application performance for service level compliance
- Maintains application availability during data set level migrations
- Allows continued growth of important or new applications
- Reduces storage platform total cost of ownership (TCO)

In addition, allocated data sets can be migrated with a dynamic swap of metadata. Table 10-1 shows the capabilities of zDMF.

<table>
<thead>
<tr>
<th>Key factors</th>
<th>Description</th>
<th>zDMF capability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic swap</td>
<td>Automatically redirects I/O to a new location after diversion completes.</td>
<td>Allows data migration without disruption.</td>
</tr>
<tr>
<td>Flexible migration options</td>
<td>Provides control over the I/O rate for data migration reads/writes.</td>
<td>Allows online migration activity while maintaining optimal application performance and service levels.</td>
</tr>
<tr>
<td>Data set grouping</td>
<td>Allows a group of data sets to be collectively migrated.</td>
<td>Provides easy management of large migrations</td>
</tr>
<tr>
<td>Robust interface</td>
<td>Fully functional ISPF windows.</td>
<td>Allows easy configuration, monitoring, and operations.</td>
</tr>
<tr>
<td>Resilient architecture</td>
<td>Current state of migration is maintained at all times, regardless of controlled or non-controlled shutdown, to ensure continued/recoverable migration.</td>
<td>Migration process is &quot;checkpoint&quot; restartable in the event of server shutdown and restart.</td>
</tr>
</tbody>
</table>

10.2 zDMF migration process

This section outlines the steps needed to conduct a successful data migration project using zDMF. zDMF is designed to move of allocated data sets that require non-disruptive data migration.
10.2.1 zDMF migration steps

After the installation, use the TSO Monitor to establish a process that performs all steps of a migration process. The sequence of steps involved in migration using zDMF is described in Table 10-2.

Table 10-2  zDMF migration steps

<table>
<thead>
<tr>
<th>Migration steps</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group definition</td>
<td>The data sets to be migrated are defined in a migration group using the TSO-based zDMF monitor.</td>
</tr>
<tr>
<td>Activating a migration group</td>
<td>Activating a migration group initiates the data migration process for that group.</td>
</tr>
<tr>
<td>Copy phase</td>
<td>Data is asynchronously copied from the source data sets to the target data sets defined in the migration group.</td>
</tr>
<tr>
<td>Synchronization phase</td>
<td>All final differences between source and target data sets in a migration group are synchronized and the group is prepared for mirroring.</td>
</tr>
<tr>
<td>Mirror phase</td>
<td>The migration group is put into a state of synchronous mirroring.</td>
</tr>
<tr>
<td>Diversion phase</td>
<td>The actual logical relocation of data sets occurs. Source and target data sets, along with their metadata, are modified and all I/O activity is redirected to the new location.</td>
</tr>
<tr>
<td>Completion phase (post migration)</td>
<td>Although the metadata has been modified, applications that were active before diversion continue to have their I/O redirected until they de-allocate the data set. An application bounce might be required in a scheduled window.</td>
</tr>
<tr>
<td>Post-Completion phase</td>
<td>After the migration is complete, the source data sets and the storage resources they are on will need to be cleaned up.</td>
</tr>
</tbody>
</table>

10.2.2 Planning the migration

Important: The planning phase is the most vital phase for a successful migration.

This section outlines the generic steps, personnel functions, and tasks that pertain to the planning phase. The high-level process consists of the following steps:

1. Establish a migration management team that consists of:
   - Primary Migration Manager
   - Alternate Migration Manager
   - Account Coordinator
   - Security Coordinator (if required for sensitive data)
   - Technical Lead Coordinator

2. Announce the migration with a notification that aligns with the application requirements.

3. Size the target configuration to match the space requirements of the source configuration.
4. Create a migration team to perform the pre-migration tasks listed in the following section. The migration team should include a technical data migration team trained to use zDMF and any other necessary utilities (that is, HSM, DFDSS, and SMS Redirection). This technical team can also act as migration mentors.

10.2.3 Pre migration tasks

Perform the following tasks before starting a migration that includes zDMF:

1. Make sure that the account coordinator identifies the owners of the data. Inform the owners about the migration and update them when required.
2. Inform the security and compliance groups about the migration if you are migrating sensitive data.
3. Use the Pre-Migration Planning Checklist to ensure that all of the pre-migration planning steps are executed.
4. Run the IDCAMS EXAMINE command with the INDEXTEST and DATATEST parameters. Verify that the source catalog has the appropriate free space and structural integrity to accommodate growth outside the scope of the migration.
5. Create a dedicated zDMF user catalog (see Example 10-1). Connect this catalog to the same systems as the source catalog. Place the catalog on a non-SMS management volume that does not contain customer data.

Example 10-1  zDMF user catalog

| Source Catalog = ICF.APPL.UCAT / Connected to System A and B |
| Target Catalog = ICF.ZDMF.UCAT / Connected to System A and B |

6. Create and relate the following aliases to the proceeding target catalog:

   Define HLQs (high-level qualifiers) that are not currently used such as X001, X002, and X003.

7. Create or add the new target volume SMS Storage Group Configuration. The new storage capacity must be the same or greater than the source storage. Consideration must also be given to any multivolume data sets.
8. Add these volumes to the appropriate storage group in ENABLE status as soon they are available.
9. Create a list of the source volumes that contain data to be moved and the associated target volumes or storage group.
10. Change the status of the source volumes to DISNEW to accommodate SMS redirection.
11. Establish a naming standard for the new target data sets (work data sets) based on the alias created in step 6.
12. Identify the time frames that you want to move between the phases. Consider DSN and Group activity, and fallback requirements when scheduling. In particular, schedule around heavy I/O loads like batch work or db2 reorganizations during the migration phase.

Remember: Some of these steps might not be applicable because of your internal policies on data migration.
10.2.4 Migration phase using DFSMS, HSM and utilities, or both

This section introduces the use of utilities to migrate data sets that are not in a persistent allocation status. You use these utilities to identify the zDMF candidates.

**DFSMS overview**

Data Facility Storage Management Subsystem (DFSMS), or SMS in short, is a software suite that automatically manages data from creation to expiration. DFSMS provides the following functions:

- Allocation control for availability and performance
- Backup/restore
- Disaster recovery services
- Space management
- Tape management

DFSMS consists of DFSMSdf (an element of z/OS) and DFSMSdss, DFSMShsm, and DFSMSrmm (features of z/OS).

Because DFSMS manages data creation, you can control the volume selection of these data sets. Use ISMF or a VARY command to establish criteria to eliminate allocation of new data sets on any volume in an SMS subsystem. This status is known as DISNEW, and disallows all new allocations.

When starting a migration that includes moving all the data sets that populate a volume, the source volumes must be placed in a DISNEW status. However, verify using ISMF that the storage pool containing the volumes has enough free space so the restriction can be enabled. Table 10-3 shows the components that address data set allocation.

<table>
<thead>
<tr>
<th>Data set components</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data class</td>
<td>Addresses the attributes of a data set.</td>
</tr>
<tr>
<td>Storage class</td>
<td>The performance level required for the data set that is in direct alignment with a specific service level agreement (SLA).</td>
</tr>
<tr>
<td>Management class</td>
<td>Addresses the appropriate backup and migration cycle of the data set.</td>
</tr>
<tr>
<td>Storage group</td>
<td>A volume or group of volumes designated for specific data sets based on predetermined criteria such as size, use, or type.</td>
</tr>
</tbody>
</table>

**SMS data redirection**

A storage subsystem that is under SMS control can direct and disable allocation to specific storage pools. In an environment that has a high delete and reallocation activity, move rate data by disabling one volume and enabling a different one in the same subsystem. This process is known as “SMS Data Redirection.”

Because SMS Data Redirection is done on a data set level, zDMF is a good solution for data sets that are not deleted on a regular basis. The first phase when migrating all the data sets in an SMS environment is to have SMS move that data using SMS Data Redirection. zDMF enables a scheduled data set movement process, which takes less time. It also allows you to use performance enhancements from the new device for those data sets not moved using SMS.
The DFSMShsm phase

Data sets that are not deleted and reallocated on a regular basis, but are not continuously allocated by an online application need special attention. Despite these data sets being frequently in use, they include a window of opportunity that can accommodate a migration. During this time, use either Hierarchical Storage Manager (DFSMShsm) or a standard utility to perform the migration.

When a new device type is replacing an old one, data sets must be moved from the old device to the new one. HSM allows you to perform such conversions of a volume with a single command. The command to convert a level 0 volume migrates all the data sets not in use on the source volume to migration level 1, then recalls them. This command allows you to clear a DASD migration volume so it can be replaced with a different type of storage device.

For example, to remove volume SG2002 from the system and replace it with a new volume, you must perform the following steps:

   - To prevent new allocations to the volume, change the status of the volume in the storage group. For example, you can establish the volume status in the storage group definition:
     
     ```
     STORAGE GROUP NAME: STGGP2
     VOLUME SERIAL NUMBER: SG2002
     SMS VOL NAME STATUS
     SG2002 ==> DISNEW
     ```

2. Move the non-allocated data sets so DFSMShsm can process from the volume using the following HSM command:
   ```
   MIGRATE VOLUME(SG2002 MIGRATE(0)) CONVERT
   ```
   DFSMShsm migrates all of the non-allocated (NOT IN USE) data sets that it can process off volume SG2002 and recalls them to other SMS-managed volumes. The DFSMSdfp automatic class selection (ACS) routine selects the volume for data set recall.

3. Remove the allocated data sets that DFSMShsm cannot process from the volume. To move any allocated data sets that DFSMShsm cannot process, use zDMF.

   ```
   Note: Data Set Services (DFSMSdss), can also be used in this phase instead of HSM. However, HSM does not require the creation of job control language (JCL).
   ```

Standard utilities

A successful migration requires a combination of various utilities that complement each other and enhance the migration process. If there is a window of opportunity to accommodate this migration, any data set migration utility, such as DFSMSdss, can be scheduled to be run.

10.2.5 zDMF migration phase

After the source configuration is placed in DISNEW, the only data sets remaining on the candidate volumes are true zDMF candidate data sets. zDMF is useful for any migration project that would normally include the use of conventional utilities requiring a planned application outage.
zDMF facilitates a scheduled data set movement process. This process provides these benefits:

- Takes less time
- Allows use of performance enhancements from the new device
- Provides a utilization rate that provides a better return on investment

**Tip:** zDMF can be used to move any data sets that are not included in the documented restriction list. The phased approach option can be used if needed due to user requirements.

### zDMF data set selection planning

zDMF includes many data set selection criteria, from a single data set to thousands of data sets and multiple groups. When using selection criteria that go beyond a fully qualified data set name, use a listing utility that uses the same selection criteria as zDMF.

The data set selection classifications can be used as selection criteria for zDMF candidates and to identify data sets to be excluded (for example, WILDCARD selection).

The selection criteria are a key factor for data set level migration and must be included in the migration planning process. Data set selection criteria can differ for each subsystem, which might require use of reporting utilities.

The following are important warnings related to this process:

- To prevent any new allocation in an SMS configuration, the SOURCE volume must be in the DISNEW status.
- Data sets that will be deleted and recreated in the zDMF migration cycle should be excluded and handled by DFSMS Automatic Class Selection (ACS) Routines.

The following are also important factors in zDMF data set selection planning:

- Review the GDG data sets for the delete activity cycle. If the cycle is shorter than the zDMF migration cycle, allow them move themselves.
- Change non-SMS-managed volumes that are mounted with a STORAGE status to PRIVATE.
- The storage subsystem must have sufficient space to accommodate normal processing. This might not be the case for data set relocation or workload balancing.
- Identify data sets with primary space 0 (SPACE=(CYL,0,100)) because they will not be migrated.
- The security profile associated with a particular data set might prevent you from moving the data set.
- When moving a data set for performance reasons, be sure that the new target volume does not already have any performance constraints.
- When doing volume consolidation, be sure that you are not selecting multiple source volumes that have performance constraints.
- Try to schedule the completion phase to take place as part of a pre-planned outage.
- Identify data sets that have a specific volser dependency. Moving data sets that have a volser dependency without including the new volser can cause a problem.
- Identify all the volumes associated with a multi-volume data set. The same number of target volumes are required.
- Use the Migration Planning Checklist to establish tasks, assignments, and status.
Restrictions
The following data sets are not supported by zDMF:

- Data sets cataloged in the master catalog or on any system resident volumes should not be included in a migration.
- Data sets used by the operating system such as LINKLIST APF authorized, page, JES2, and JES3. These data sets are not supported because they might not divert properly, causing serious system outages.
- Catalogs.
- ISAM.
- Individual PDS members.
- HFS data sets.
- Page and swap data sets.
- Temporary (&&) data sets.
- Undefined data sets.
- Data sets allocated with no primary space or utilized space.
- Volume-specific data sets, including volume table of contents (VTOC), VSAM volume data set (VVDS), and volume table of contents index (VTOCIX).

Both control unit types must be of an equal or higher type.

Example:
2105 -> 3990 will not be moved
3990 -> 2105/2107 will be moved
2105 -> 2105/2107 will be moved

The following data sets are flagged and do not go into Diversion phase until all associated applications are terminated on all participating systems. For more information, see the “Migration scheduling techniques” section in the Installation and Reference Manual.

- VSAM record level sharing data sets
- BDAM data sets that require absolute track

Data migration classifications
The following are the classifications used with data migration:

- Manage the use of source DSN masks to ensure that the population of data sets to be migrated by any one group meet the following limitations:
  - Less than 1,500 data sets
  - Less than 40,000 extents total
- Appropriately sizing a migration group reduces exposures related to ECSA, CSA, and extended private storage consumption. It also mitigates the ENQ impact of migration processing on catalogs and volumes.

Important: When using TDMF or zDMF, make sure that catalog volumes are moved in a separate window. In addition, make sure that catalog maintenance is not scheduled for a basic catalog structure (BCS) associated with data sets that are being moved.
An EXCLUDE LIST to separate data sets into various groups. Create a MASTER
EXCLUDE LIST. This list includes identifying data sets that are:
- RESTRICTED (ISAM, HFS, and so on)
- VOL SER DEPENDANT data sets
- SOFTWARE/SYSTEM data sets that are sensitive to movement

Migration group standards
Follow these guidelines when creating migration groups:
- Create group naming standards that provide easy identification. zDMF appends a ZD to a
  member name, limiting your choice to six characters.
- Create groups that you can manage based on the number of data sets.
- Retain PDS members for reference.

Migration scheduling techniques
zDMF provides capabilities that are imperative for current mainframe processing
environments:
- Non-disruptive migration capabilities
- Flexibility

With zDMF, you can respond immediately to a storage-related issue. In addition, you can
manage your migration so that you maintain data set level mirroring until one of the following
phases:
- The Diversion and Completion phases in the same window
- The Completion phase at a later time after the Diversion phase

These phases are known as Extended Mirrored phase and Non-Extended Mirrored phase.
Additionally, zDMF can suspend and resume a migration using the ISPF interface. For more
information, see the zDMF Installation Manual.

Extended Mirrored phase
The zDMF migration phases from the Activation phase to the MIRRORING phase do not
include catalog alteration. While in the Mirrored Phase the two cataloged data sets, target and
source, exist and all the I/O is directed to the original source data set. Therefore, you can stay
in the Mirrored phase to accommodate the following needs:
- Migrations that include a scheduled outage that provides enough time for the Diversion
  and Completion phases
- To align with internal change control policies
- For sensitive data sets that are controlled by external compliance regulations
- To accommodate business unit and or customer requirements

Extended Mirrored phase and EDSC
The EDSC option allows you to automatically divert any data sets in the group that are not
currently allocated immediately after the group enters Mirror state. The EDSC option includes
these settings:
**EDSC No:** All data sets remain in Mirror state until you issue the DIVERT command for the
group.
**EDSC Yes:** Non-allocated data sets are diverted automatically immediately after
synchronization. They are then eligible for completion processing as though the
group itself is diverted.
**Non-Extended Mirrored phase**

The zDMF migration management process can be set to go from the Activation phase to the Completion (post migration) phase without extending the Mirrored phase. This process does include catalog alteration, diverts all the I/O to the new target data set location.

Staying in the Non-Extended Mirrored phase can be used to accommodate the following needs:

- Migrations that require immediate relocation of a data set or data sets for performance improvements.

  **Tip:** When moving a data set for performance reasons, be sure that the new target volume does not already have a performance problem.

- Technology refresh projects that include devices of larger or smaller capacity.
- Physical separation of sensitive data.
- To accommodate business unit and or customer requirements.
- To accommodate storage-related issues immediately.

### 10.2.6 Typical migration process scenario

This section outlines the steps of a typical storage migration. In this scenario, a technology upgrade is required because the lease of the current storage subsystem is ending. Therefore, data needs to be migrated from old to new storage. While performing this conversion, it was also decided to install new higher capacity drives.

The migration metrics are shown in Table 10-4.

<table>
<thead>
<tr>
<th>Old storage</th>
<th>New storage</th>
<th>Capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>800 MOD-3 to</td>
<td>800 MOD-3 to</td>
<td>2270 GB</td>
</tr>
<tr>
<td>120 MOD-3 to</td>
<td>40 MOD-9</td>
<td>340 GB</td>
</tr>
<tr>
<td>200 MOD-9 to</td>
<td>200 MOD-9</td>
<td>1702 GB</td>
</tr>
<tr>
<td>240 MOD-9 to</td>
<td>80 MOD-27</td>
<td>2043 GB</td>
</tr>
<tr>
<td>1360 volumes to</td>
<td>1120 volumes</td>
<td>TOTAL=6355 GB</td>
</tr>
</tbody>
</table>

The new subsystem must be installed. In this scenario, the old subsystem must be removed within 30 days of the arrival of the new subsystem to avoid costly storage overlap. It was quickly decided that conventional disk-to-disk copy was not feasible because of the long application outage window required. Also, the hardware migration utility of the vendor was not usable because of compatibility issues going from ESCON to FICON. Ultimately, a combination of host-based utilities and products was chosen to perform the migration.

Because the environment consisted of 80-90% SMS, the operations staff felt they could take advantage of SMS redirection and the following functions:

- DFDSS to copy non-allocated files
- TDMF for volume level migrations
- zDMF for allocated files
Selection logic
TDMF was chosen to move the volume data. However, a non-disruptive logical data set mover was needed to accommodate migration requirements. TDMF was in use but not zDMF. It was determined that both TDMF and zDMF can accommodate the migration needs. These two products provide for vendor-neutral, non-disruptive migration and came with a strong endorsement from the storage vendor.

Running the migration
It was decided to perform the most complex migration, migrating the smaller to larger volumes, first. The migration involved the following steps:

- ICKDSF Minimal Init
- Migrating smaller to larger volumes with TDMF
- Migrating using SMS and HSM processes
- Migrating one-for-one volumes using TDMF
- Example on page 444
- Completing the migration
- Verifying the migration

ICKDSF Minimal Init
Run an ICKDSF Minimal Init with a VOLID like XXucb#. This function was used rather than initializing the new volumes with VOLIDs that corresponded to their naming convention. In addition, they were not included in the correct SMS storage pool. No SMS updates are done to include these volumes in the SMS pools.

Migrating smaller to larger volumes with TDMF
TDMF was used to migrate the smaller volume to a larger volume, that is, one MOD-9 to an MOD-27. This completed one-third of the migration that required resizing without any disruption to production. It took less than one hour to migrate 40 MOD-3 to MOD-9 volumes, and about four hours to migrate 80 MOD-9 to MOD-27 volumes.

Because the source VOLID was being copied to the target volume, no SMS changes were needed. Changes were unnecessary because the volume was already part of the wanted SMS rules. Also, the VOLID naming standard is propagated to the new volume.

The TDMF migration was set up to dynamically start ICKDSF to reformat and expand the VTOC of a volume. This function is performed when the source VTOC characteristics do not match that of the target device.

Execute this step if you have correct settings for your VTOC size when volume was moved to new target volume. You can also do it if TDMF (ICKDSF) can extend all volumes to a correct VTOC size.

Use the ICKDSF EXTVTOC option only if migrating from one size device to another, or if a volume was migrated and no REFVTOC was performed. Only indexed VTOCs are extended. Non-indexed VTOCs, including volumes with damaged indexes, are allowed.

You can specify a specific number of tracks you want the VTOC to be or you can allow TDMF use its own algorithm as follows. The minimum size of the new VTOC is the greater of the current VTOC size on the source and target. The VTOC can be extended further depending on the number of data sets on the volume:

- If the volume is less than half full, the VTOC is extended to contain the current number of data sets multiplied by the ratio of target to source device size, plus 25%.
- If the volume is more than half full, the VTOC is expanded to handle the situation where the target volume is full of data sets with the same average size.
If the index needs to be extended, but the VTOC does not, TDMF attempts to extend the VTOC by one track. If the VTOC cannot be extended because there is a data set next to it, REFVTOC runs unless there is insufficient space in the index for the additional VPSMS.

**Important:** Resizing works only if free space is available directly behind the VTOC. If there is no free space VTOC cannot be resized, but the new number of cylinders are updated in VTOC.

*Migrating using SMS and HSM processes*

The remaining two-thirds of the source volumes were consolidated to larger capacity volumes. These volumes were placed in the SMS DISNEW status. This status automatically migrates files that were deleted and reallocated during daily and weekly production runs to the new devices. It also prevents SMS from allocating any new files or extents on the old volumes.

Operations staff can go on to their other migration tasks and allow the normal SMS routines to move the files. However, they did need to revisit this task to determine which files SMS did not migrate.

HSM can also be used to migrate those data sets that are not in use using the HSM command MIGRATE VOLUME(SG2002 MIGRATE(0)) CONVERT.

**Tip:** Using HSM would eliminate the creation of JCL to accommodate the migration.

*Migrating one-for-one volumes using TDMF*

Operations staff used TDMF to migrate the remaining disks that were one-for-one. These disks included 800 MOD-3 to MOD-3 and 200 MOD-9 to MOD-9 DASD. These volumes also include all volumes with data sets pointed out in the restriction list of zDMF.

*Migrating using zDMF*

The only migration that remains to be performed is for the files which are not reallocated by SMS to higher capacity disks in Steps 1-4. This migration was completed by zDMF.

The files that remained were files that had not been through a processing cycle. These included the following kinds of files:
- Weekly job runs
- Files that had been created a long time ago and never got deleted
- Files that were constantly in use such as the DB2 and CICS files

Consideration was given to using DFDSS to migrate the remaining files not in allocation. It was determined that using zDMF, operations staff can accomplish the same thing and also handle data sets that were in use at the time. In addition, zDMF provided the flexibility of allowing data sets to go through allocation during this process. DFDSS, however, would have periodic allocation issues as production continued to run.

*Completing the migration*

All files except the ones with persistent allocations were migrated to completion. The files that remained in zDMF Diversion mode were identified using the TSO Monitor. Those applications were then scheduled to be bounced over the weekend.

After zDMF completed all data set migrations, the Storage Migration project to the new technology was complete.
Verifying the migration
An additional day was spent to verify that all data had migrated as planned and post-migration
tasks had been completed.

Migration time line estimate
The following is an estimated time line (see Figure 10-1) for the migration described
previously:

- Pre-migration: Install new storage the first week.
- Migration Step 1: ICKDSF INIT new storage.
- Migration Step 2: TDMF small volumes to large volumes.
- Migration Step 3: SMS DISNEW redirection.
- Migration Step 4: TDMF equal size volumes.
- Migration Step 5: zDMF small to large volume “files” (files not moved by SMS redirection).
- Migration Step 6: Scheduled application bounce (if required).
- Migration Step 7: Clean-up.
- Post-migration: Two weeks to remove old storage (installation, migration, and
de-installation complete in one month).

Figure 10-1  Sample migration time line

10.2.7 Migration performance and scalability
This section addresses the zDMF performance parameters that establish the user needs with
specific migration requirements and system capabilities. Guidelines for sizing an zDMF
migration group are also included.

zDMF performance considerations
zDMF uses both asynchronous and synchronous techniques to copy data from source
locations to target locations, minimizing performance impact.

For performance reasons, do not place hlq.HGZD325.GZDLLIB library in the system link list
(LNKLSTxx). If the library is placed in the system link list (LNKLSTxx), remove the STEPLIB
DD statement from the JCL before running zDMF. If you place hlq.HGZD325.GZDLLIB in
PROGxx or IEAAPFx of SYS1.PARMLIB, use the STEPLIB DD statement.

The zDMF server must always be active and able to immediately process requests. zDMF
requires a high execution priority, and should be put in a response-oriented performance
group. The zDMF-server-started task JCL specifies the configuration PDS member that
contains the startup parameters. The zDMF server configuration PDS member is located in hlq.HGZD325.SAMPLIB(CONFIG).

Generally, do not change default values in CONFIG member. However, you might be directed to do so by support in some special cases.

**zDMF parameters**
The hlq.HGZD325.SAMPLIB(CONFIG) member has the following parameters:

- **MAXIO**: MAXIO determines the maximum overall number of I/O requests that can be active at one time on the server. Because I/O buffers and control areas are allocated based on the MAXIO value, select a number appropriate to the resources available.

  Approximately 1 MB of memory is allocated for each integer you add to the MAXIO specification. For example, you select a MAXIO value of 25, approximately 25 MB is allocated. The memory allocation is fixed during active I/O.

  \[
  \text{MAXIO} = \text{number}
  \]

  Where \textit{number} is the maximum number of Overall Requests that can be active at one time on the server. The minimum value is 0. If you select 0 or do not specify a MAXIO value, the number of Overall Requests defaults to 25.

  - Default: \text{MAXIO}=25

- **MAX_CHANNEL_IO**: Specifies the maximum number of concurrent I/O requests that can be issued to a channel during the Copy phase or Mirror Synchronization phase. The MAX_CHANNEL_IO limit applies to any active I/O against a channel group, whether read or write. If the source and target devices are on the same channel group, the MAX_CHANNEL_IO limits the total concurrent I/O requests on the channel.

  \[
  \text{MAX\_CHANNEL\_IO} = \text{Requests}
  \]

  Where \textit{Requests} is the maximum number of concurrent I/O requests. The smallest value is 0. There is no theoretical maximum value. However, the largest practical value is the current MAXIO value.

  - Default: \text{MAX\_CHANNEL\_IO}=15

- **MAX_DEVICE_IO**: Specifies the maximum number of concurrent I/O requests that can be issued to devices containing a data set migration pair. This limitation applies during the Copy and Mirror Synchronization phases.

  \[
  \text{MAX\_DEVICE\_IO} = \text{Requests}
  \]

  Where \textit{Requests} is the maximum number of concurrent I/O requests. A value of 1 to 5 can be specified.

  - Default: \text{MAX\_DEVICE\_IO}=3

- **MAXTRK**: This optional parameter specifies the size of I/O operations in tracks that zDMF I/O copy operations use to transfer less than a full cylinder (one extent) of data. The MAXTRK value is used to reduce the application response time impact of zDMF Copy operations immediately following activation. For example, MAXTRK=5 causes zDMF to move one extent in three I/O operations. Splitting the extent reduces the time the device is unavailable to application I/O operations into three short windows.

  \[
  \text{MAXTRK} = n
  \]

  Where \textit{n} is a value from 1 to 15.

  - Example: \text{MAXTRK}=5
  - Default: \text{MAXTRK} is not used in CONFIG (parameter marked as a comment)
UNIDENTIFIED_SYSTEMS_ACTION: Tells zDMF to check that a zDMF Server is active on all z/OS images that can access a shared migration volume on a standard 3990-6 compatible control unit. The possible values are:
- IGNORE causes no action
- WARNing results in warning messages (GZD4192W or GZD4194W)
- TERMinate causes the group activation to fail if all z/OS images with access to the source volume do not have an active zDMF Server. The failure message is GZD4191E or GZD4193E. If a target volume fails the check, it is removed from candidacy, message GZD4195I is issued, and the activation continues as long as other target volumes are available.

Tip: Initially run in TERMINATE mode to verify correct operations and build an IGNORE_SYSTEMS list if needed.

This parameter works with 3990-6 control units or later (2105/2107) only. TDMF cannot check attached systems with 3990-3 (old HW).

Important: You are responsible for running the correct number of Started Tasks.

Sizing an zDMF migration group
Serialize state changes when multiple groups are used to migrate a population of data sets. To avoid possible application performance impact and deadlock exposure, ensure groups progress through the Activation and Pending-Divert states in serial fashion. For activations, ensure that each group is showing a small percent of Copy/Synchronization completed before activating successive groups. Likewise, ensure that each group is in a state of Mirrored before entering the divert command to put successive groups in diversion.

Application performance impact can be reduced by use of the MAXTRK zDMF server parameter.

The MAXTRK parameter causes zDMF I/O operations to be performed using less than one cylinder. For more information, see the “Configuring the zDMF Server Parameters” in the zDMF Installation and Reference Guide.

zDMF storage requirements
zDMF checks the available free space in ECSA while the group is activated. If there is not enough free space available, the group are not activated. The basic ECSA, CSA, and Extended Private storage requirements for zDMF are as follows:
- A global area is acquired that needs X'698' (1,688) bytes of fixed ECSA.
- Operating system interface modules are loaded into fixed ECSA, and require at least X'DAA08' bytes (874.5 KB).
- The PC routine is loaded into pageable ECSA, which currently requires X'B3F00' bytes (720 KB).
- The storage pools (all fixed ECSA) are initialized with X'480000' bytes (4.5 MB). When groups are mirroring or diverting, the storage pools expand depending on the complexity of the channel programs.
- When a group is promoted and activated, the parsing routine uses ECSA to store images.
- For each active group, the following are true (in fixed ECSA):
  Control blocks for which space is not freed:
– One DWGRP block, X’A8’ (168) bytes, until the next IPL, or the product is shut down.
– One DWGDEV block for each source and target device, X’F8’ (248) bytes.
– A sparse array for extent look-up from each device block. Minimum size is X’400’ bytes (1 KB).

Control blocks for which space is freed when the group completes:
– One QCDSN block for every data set and each individual VSAM component or AIX, X’2B8’ (696) bytes.
– One DWEXT block for each data set extent, X’68’ (104) bytes.

Note: At least 0.85 MB is resident in fixed ECSA until the next IPL for all components, except for the PC routine and possibly the storage pools.

Security
Protect zDMF using IBM RACF®, ACF2, or other security utilities to prevent use by unauthorized personnel. Use of zDMF by unauthorized personnel might result in the inappropriate transfer of a data set out of a specified isolated environment.

Volume checklist
Be sure that the TARGET configuration is added to the appropriate storage group and all the volumes remain in DISALL status until you are ready to start the migration. The volume status then changes to ENABLE.

Be sure that all of the TARGET volumes meet the following requirements:

➤ Are initialized for SMS (ICKDSF STORAGEGROUP)
➤ Contain a configuration large enough for a VVDS
➤ Contain an INDEX VTOC

Because the INDEX VTOC is required on an SMS-managed volume, ISMF can be used to identify a disabled or broken INDEX VTOC.

Be sure that all the SOURCE volumes are in DISNEW status. This includes volumes with a similar capacity configuration even if they do not contain data sets that are scheduled to be moved. An example is all mod 3 in the same STG GRP. In an SMS configuration, this status prevents any movement back to a DEVICE TYPE that is scheduled for removal. If DISNEW status was enabled using an SMS VARY command, the next ACS TRANSLATE and VALIDATE can reverse that status.

Tip: SNAP requires an INDEX VTOC on the volume. Also, the VTOCIX must match the volser or the migration fails.

In this example, the migration continues:
SYS1.VTOCIX.TD538D at volume TD538D

In this example, the migration fails:
SYS1.VTOCIX.SPMS19 at volume YPMS19

Migrating software vendor data sets
The zDMF data set selection criteria and other restrictions are based on specific access methods and data set types. These restrictions are clearly outlined in the zDMF product documentation.
However, you must first consider additional software vendor restrictions on some product data sets. These restrictions are essential to preserve data integrity and product-related functionality. Additionally, before running the zDMF migration, make sure that you have documentation of user exits, product customization, and cross-application dependencies unique to your environment.

zDMF can move any data sets that conform to supported data set lists and other restrictions. However, consider additional application restrictions and consult with the person responsible for the application. Also, keep in mind that each installation, although apparently similar, can have subtle differences that might jeopardize the migration.

### 10.2.8 Post-migration

The creation of a separate user catalog for the zDMF target data sets expedites the post-migration cleanup process. Additionally, the renamed source data set provides content verification that you can use for a post-migration audit trail. The following are the steps for various post-migration scenarios:

For a full volume migration:
- Delete the aliases related to the zDMF user catalog.
- Delete the zDMF user catalog.
- Initialize the old source volumes.

For a selective data set migration:
- Delete the work data sets using IDCAMS:
  
  ```
  DELETE (X00x.*)
  ```
- Delete the data sets using ISPF 3.4 or ISMF.
  Or use the zDMF monitor function Z Generate Source DSN Cleanup JCL in front of a completed group delete all work data sets.
- Delete the aliases related to the zDMF user catalog.
- Delete the zDMF user catalog.

For a full volume migration with post migration verification:
- List the original zDMF candidate data sets that can be found on the original volumes.
  If the volume contained data sets that are either restricted or invalid, they must be factored in the original data set count.
- Based on the zDMF selection criteria, list the new data sets using IDCAMS.
- Compare the data sets. Because both the target and source data set configurations consist of cataloged data sets, the contents of either can also be verified.

For a data set migration with post-migration verification, perform the following steps:
- List the original zDMF candidate data sets using any listing utility and the same zDMF selection criteria.
  If the volume contained data sets that are either restricted or invalid, they must be factored into the original data set count.
- Based on the zDMF selection criteria, list the new data sets using IDCAMS.
- Compare the data sets. Because both the target and source data set configurations consist of cataloged data sets, the contents of either can also be verified.
As part of the post-migration activities, address the following concerns:

- Create a clean-up schedule to determine how long you want to keep the zDMF target data sets.
- Determine how you want to treat the zDMF data sets:
  - Back up before delete
  - Delete without backup
  - Delete the BCS and Init the volumes
  - Content verification for post-migration audit

10.3 Important zDMF-related topics

This section includes other important concepts that you need to understand in the context of an zDMF migration.

10.3.1 Understanding the catalog structure

Because zDMF is a data set-level migration tool that includes updating the information in the ICF catalog, you must have a basic understanding of the ICF catalog structure. This section provides a high-level outline of the structure to help engineers do the following tasks:

- Perform diagnostic tests for the pre-migration process
- Identify the cause of migration problems resulting from erroneous entries

The ICF catalog records are stored in two components:

- The basic catalog structure (BCS): BCS is considered the catalog. The BCS is a VSAM key-sequenced data set (KSDS). Its primary function is to point to the volumes on which a data set is located. The BCS is created when either a user or master catalog is defined using access method services (AMS). A BCS does not have to be on the same volume as the data set it references. There can be more than one BCS on a volume.

- The VSAM volume data set (VVDS): VVDS is considered an extension of the VTOC. The VVDS is a VSAM entry-sequenced data set (ESDS). It contains the information required to process VSAM data sets and, in a Storage Management Subsystem (SMS) environment, it contains the volume-related information for non-VSAM SMS-managed data sets. There is one VVDS on each DASD volume that contains a VSAM or SMS-managed data set cataloged in an ICF catalog. The VVDS is always on the same volume as the data set it references.

The relationship between the BCS and the VVDS is many-to-many. A BCS can point to multiple VVDSs, and a VVDS can point to multiple BCSs.
Any volume containing a BCS also contains a VVDS because the BCS is itself a VSAM data set.

The records in both the VVDS and BCS are made up of variable-length cells and subcells. The two cell types that are often referred to are the VSAM volume record (VVR) and the non-VSAM volume record (NVR), which are both held in the VVDS. The VVR contains information relating to VSAM data sets. The NVR contains information relating to non-VSAM SMS-managed data sets.

Most data sets have entries in only one VVDS. However, multivolume data sets have entries in the VVDS of each volume they are allocated to (type Q for a secondary record).

Table 10-5 shows the structure of the ICF catalog.

### Table 10-5  ICF catalog structure

<table>
<thead>
<tr>
<th>Information</th>
<th>VSAM data set</th>
<th>SMS-managed non-VSAM data set</th>
<th>Non-SMS-managed non-VSAM data set</th>
<th>Uncataloged non-VSAM data set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volume</td>
<td>BCS</td>
<td>BCS</td>
<td>BCS</td>
<td>n/a</td>
</tr>
<tr>
<td>Data set type</td>
<td>BCS</td>
<td>BCS</td>
<td>BCS</td>
<td>n/a</td>
</tr>
<tr>
<td>Association</td>
<td>BCS</td>
<td>BCS</td>
<td>BCS</td>
<td>n/a</td>
</tr>
<tr>
<td>Ownership</td>
<td>BCS &amp; VVDS</td>
<td>BCS &amp; VVDS</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>SMS class info</td>
<td>VVDS</td>
<td>VVDS &amp; VTOC</td>
<td>VTOC</td>
<td>VTOC</td>
</tr>
<tr>
<td>Data set attributes</td>
<td>VVDS &amp; VTOC</td>
<td>VTOC</td>
<td>VTOC</td>
<td>VTOC</td>
</tr>
<tr>
<td>Extent description</td>
<td>VVDS</td>
<td>VTOC</td>
<td>VTOC</td>
<td>VTOC</td>
</tr>
<tr>
<td>Catalog name</td>
<td>VVDS</td>
<td>VVDS</td>
<td>n/a</td>
<td>n/a</td>
</tr>
</tbody>
</table>

### 10.3.2 BCS record types

The first cell in each record has a cell type field, which is also the record type (or ID). The record type can be identified using the DIAGNOSE command. The following are the possible record types and their one-character identifiers.

BCS ID record types:

- A non-VSAM data set
- B generation data group
- C cluster
- D data component
- E VSAM extension record
- G Alternate index
- H GDS
- I index component of a cluster
- J GDG extension cell
- L library
- R path
- T true name
- U user catalog connector
- W volume
- X alias

VSAM Volume Record (VVR):
- Z primary record
- Q secondary record and data sets allocated with IMBED
- N Non-VSAM Record (NVR) for non-VSAM data sets

10.3.3 Catalog diagnostic recommendations

The BCS, VVDS, and VTOC each supply a portion of the information required to process a data set. Damage to any piece or a mismatch between them can prevent you from accessing your data even if there is no problem with the data set itself. Most data sets have entries in only one VVDS or VTOC. The exception is a multivolume data set, which has entries in the VVDS or VTOC of each volume it occupies.

Because the zDMF data movement process includes catalog alteration, you must start with a functional catalog. In addition, verify that the candidate data sets are correctly cataloged on the system or systems to which zDMF requires access. Use SSMzOS or IDCAMS to perform the required diagnostic tests.

BCS/VVDS/VTOC synchronization

A SSMzOS catalog diagnostic checks the following items:
- Data structure
- Index integrity
- BCS/VVDS/VTOC synchronization

The catalog diagnostic tests can also generate IDCAMS control cards to fix the errors detected. Specifically, the IDCAMS control cards for DEFINE RECATALOG can be generated to recreate missing VSAM and non-VSAM catalog entries. The catalog commands and their actions are shown in Table 10-6.

Additionally, there are diagnostic routines to detect and remove superfluous catalog entries. These diagnostic routines can be directed to your entire catalog environment, or assigned to a specific catalog, volume, or object (catalog, VVDS, or volume). Specific objects are selected using keywords that allow specific and generic parameters. When SSMzOS catalog-related functions are run on a scheduled basis, these steps establish consistent catalog synchronization. This synchronization is important when running post-catalog recovery diagnostic tests and explaining erroneous entries in a BCS or a VVDS.

Table 10-6 Catalog command and action summary table

<table>
<thead>
<tr>
<th>Command</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELETE NOSCRATCH</td>
<td>Removes only BCS entries. VVDS and VTOC are not affected.</td>
</tr>
<tr>
<td>DELETE TRUENAME</td>
<td>Removes unrelated truename record if the associated cluster record does not exist.</td>
</tr>
<tr>
<td>DELETE VVR or DELETE NVR</td>
<td>Removes unrelated VVR or NVR. Also removes VTOC entry, if present.</td>
</tr>
<tr>
<td>DEFINE RECATALOG</td>
<td>Creates a BCS entry pointing to existing VVRs or NVR only in the BCS they name.</td>
</tr>
</tbody>
</table>
The EXAMINE command
To guarantee that a BCS and an associated backup are structurally sound, the IDCAMS EXAMINE command must be included in the scheduled BCS backup. This command traces failures caused by unsynchronized entries in the catalog, and shows if the catalog might be structurally unsound. Structural problems affect BCS characteristics as a VSAM key-sequenced data set, not as a catalog. Test both the index and data components of a BCS, and identify the amount of free space remaining in the BCS.

Sample JCL
This section lists sample JCLs.

Diagnostics: BCS/VVDS/VTOC synchronization
The following JCL provides VVDS to BCS synchronization:

```
//STEP01 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//VOLDD DD VOL=SER=PROD01,DSN=SYS1.VVDS.VPROD01,NAME,DISP=OLD
//SYIN DD *
DIAGNOSE VVDS INFILE(VOLDD) COMPAREDS(ICF,BCS.UCAT)
ERRORLIMIT(200) INDEXTEST DATATEST
/*

For VTOC to BCS synchronization, see the SSMzOS catalogs functions.

Diagnostics: IDCAMS DIAGNOSE
This example shows how to determine whether the index component of your catalog has structural errors:

```
//EXAMEX1 JOB
//STEP1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//SYIN DD *
EXAMINE NAME(ICFCAT.V338001/MASTRPW) -
ERRORLIMIT(200) INDEXTEST DATATEST
/*

The EXAMINE command is used to analyze the index component of an integrated catalog facility catalog. Its parameters are:

- **NAME**: Specifies the catalog name and its master password. The catalog must be connected to the master catalog.
- **INDEXTEST**: Specified by default.
- **ERRORLIMIT(0)**: Suppresses the printing of detailed error messages.

Sample output:

```
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

Then following is an example of the IDCAMS control statements that can be used for VVDS and BCS diagnostics.

Compares the forward pointers from the BCS to the VVDS:

```
DEFINE NONVSAM
```

Creates BCS entry pointing to non-system-managed non-VSAM data set. This command does not verify the existence of a data set on the named volume.
10.3.4 Using Global Resource Serialization (GRS)

In a Global Resource Serialization complex, programs can serialize access to data sets on shared DASD volumes at the data set level. A program on one system can access a data set on a shared volume while other programs on any system can access other data sets on the same volume. Serialization can therefore reduce contention for these resources and minimize the chance of an interlock occurring between systems. This serialization ends the need to protect resources by job scheduling.

Because global resource serialization maintains information about global resources in system storage, data integrity is maintained during a system reset while a reserve exists. A global resource serialization complex also allows serialization of shared logical resources. These resources do not have to be directly associated with a data set or DASD volumes. An ENQ with a scope of SYSTEMS can be used to synchronize processing in multisystem applications. Combining the systems that access shared resources into a global resource serialization complex can solve the problems related to using the Reserve macro.

How global resource serialization works depends on the z/OS operating system level. It also depends on whether the cross-system coupling facility (XCF) environment the systems are running in is sysplex or non-sysplex.

The complex consists of one or more systems:

- Connected in a ring configuration through:
  - Global Resource Serialization-managed channel-to-channel (CTCs) adapters
  - XCF communication paths (CTCs)
  - Signaling paths through a Coupling Facility
- Connected to a coupling facility lock structure in a star configuration through:
  - Signaling paths
  - XCF communication paths
  - A combination of both signaling paths and XCF communication paths

No matter which configuration you choose, global resource serialization processing (ISGENQ, ENQ, DEQ, and RESERVE requests) for resources is the same.

10.3.5 Ring complex

As stated earlier, a global resource serialization ring complex consists of one or more systems connected by communication links. Global resource serialization uses the links to pass information about requests for global resources from one system in the complex to another.

The global resource serialization complex consists of every system that indicates, at IPL time, that it is to be part of the complex. This complex is not affected by the physical configuration of systems and links. The systems in the complex might not all be actively using global resource serialization at any particular time. Those systems that are actively using global resource serialization to serialize access to global resources make up the global resource serialization ring.
Figure 10-2 shows a four-system global resource serialization ring complex. When all four systems in the complex are actively using global resource serialization, the complex and the ring are the same. The complex has a communication link between each system and every other system. It is therefore a fully connected complex.

A sysplex requires full connectivity between systems. Therefore, when the sysplex and the complex are the same, the complex has full connectivity. Although a mixed complex might not be fully connected, a fully connected complex allows the systems to build the ring in any order. It also allows any system to withdraw from the ring without affecting the other systems. The complex offers more options for recovery if a failure disrupts ring processing.

For example, if system System1 in Figure 10-2 were to fail and end its active participation in serializing access to global resources, it would still be part of the complex. However, it would not be part of the ring.
10.3.6 Star complex

The star method for processing requests for global resources operates in a sysplex like any other z/OS component that uses the Coupling Facility (Figure 10-3). Unlike the ring method, which might require you to set up XCF signaling paths for the SYSGRS group, the star method operates well without any special definitions. There is no need to set up XCF signaling paths between systems or to define CTC links directly to global resource serialization.

![Figure 10-3 Star concept](image)

XCF requires a DASD data set, called a sysplex couple data set. This data set is shared by all systems. An alternate data set can be used to facilitate migration from a ring to a star complex. On the sysplex couple data set, z/OS stores information related to the sysplex systems, and XCF groups such as global resource serialization. The following policies are used to manage global resource serialization:

- Coupling facility resource management (CFRM) policy, which is required, defines how MVS manages Coupling Facility resources.
- Sysplex failure management (SFM) policy, which is optional, defines how MVS is to manage the system. It signals connectivity failures and IBM PR/SM™ reconfiguration actions. Generally, use the SFM policy.

**Tip:** XCF is a component of the z/OS operating system that supports cooperation between authorized programs running in a sysplex.

10.3.7 Reserve handling requirements

zDMF has specific requirements in the area of reserve handling. More specifically, zDMF requires that all hardware reserves issued by z/OS address spaces be converted to globally propagated ENQ requests. These reserves include system, application, and jobs.
Understanding Global Resource Serialization and Hardware Reserves

Global resource serialization (GRS) is a function that allows serialized use of resources. Through enqueue and dequeue facilities, usage of shared resources can be guaranteed. Resources can either be simultaneously shared across multiple users of the resource, or granted exclusively to one user.

GRS is useful in a shared storage environment in which multiple z/OS images are using the same storage resources. To control use of a resource, the z/OS operating system issues a hardware reserve to the physical device. This reserve ensures that no other z/OS image in the shared storage environment can access that device so long as the reserve is still held.

To improve system performance and throughput, products that convert hardware reserves to globally propagated ENQ requests were developed. An example is the Multi-Image Manager by Computer Associates. These products communicate with all systems in the shared storage complex. They ensure exclusive usage by issuing a software enqueue on all systems in the shared storage environment.

When using zDMF to migrate data, the following protections are provided while converting hardware reserves to globally propagated ENQ requests:

- Ensuring the data integrity of data for data sets being migrated
- Avoiding deadly embraces in multisystem environments

Ensuring data integrity for data sets being migrated

To ensure the data integrity of data sets being migrated, hardware reserves must be converted to globally propagated ENQ requests. In the following examples, data set DS1 is being migrated from source Volume A to target Volume B. The applications are running on two z/OS servers, MVSA and MVSB, and both require exclusive serialization of the resource.

**zDMF impact on serialization before diversion**

zDMF phases that run on a migration before diversion have no impact on resource serialization. These phases include Group Definition, Activation, Synchronization, and Mirroring. Because all applications are continuing to read and write data from the source location, all serialization is targeted to the source location. For example, data set DS1 on Volume A has all reserves issued to that location across all applications and systems for serialization of the resource. Therefore, all I/O activity to the Mirrored data set or target on Volume B is also serialized.

**zDMF impact on serialization after diversion**

During the zDMF diversion phase, hardware reserves are converted to ENQ requests to ensure data integrity of data sets being migrated.

After entering the diversion phase, applications that remain active after diversion will continue to serialize usage of the resource as though it still on the source volume. Applications starting after diversion are allocated directly to the new target location and do not require any I/O diversion activity by zDMF. These applications do not require alteration because all catalog and volume level metadata are modified to reflect the new location of the data set.

When this scenario occurs, an exposure exists with resource serialization for the data set. The following is an example of how this scenario can occur:

- MVSA reserves Volume A for data set DS1 which is in diversion, reads data, and prepares to write it back. The data read occurs on Volume B because the migration is in the Diversion phase.
- On MVSB, a new application starts and reserves volume B. The application reads data from data set DS1, modifies it, and writes it back to data set DS1.
During this time, the zDMF channel program for MVSA must wait because Volume B is reserved to MVSB. After MVSB releases the reserve on Volume B, MVSA will overwrite what MVSB wrote, and then release the reserve on Volume A.

In this scenario, a classic data integrity exposure occurs. The data is overwritten without any indication that this has occurred.

If the reserve is converted to a global enqueue, the resource is protected by name and remains secure. This protection remains as long as all applications protecting any associated resource use the same name and do not rely on the actual device reserve for serialization. An example is when other data sets are thought to be on the same volume or any logical resource. If the reserve is not converted, the benefit of reserve protection is lost between pre- and post-diversion applications.

Avoiding deadly embraces in a multi-system environment

The SYSVTOC and SYSZVVDS resource names are the QNAMES for resource serialization of the VTOC and VVDS of each volume. These data sets contain all the control information for data sets on that volume.

Data sets grow and shrink with usage. As they do, the VTOC and VVDS must be updated to reflect the space usage on that volume. zDMF mirrors all activity for a data set on a source volume to a target volume. When extent changes occur, zDMF must ensure that similar changes occur on the target volume. For example, if DS1 (data set 1) on Volume A adds an extend, zDMF must also add an extent on Volume B for the mirrored data set.

If hardware reserves are not converted for the VTOC and VVDS, hardware reserves must be issued to both physical volumes. This can result in a deadly embrace.

To avoid this situation, these resources must be converted from hardware reserves to ENQ requests. This conversion allows special handling code in zDMF to prevent multiple systems from trying to exclusively acquire the same volume resources.

If these resources are not converted from hardware to ENQ requests, you might have to cancel address spaces to eliminate the deadly embrace.

To convert the reserves, use products such as CA-MIM or GRS by IBM. zDMF provides a stand-alone utility called Reserve Monitor. This utility detects hardware reserves and determines whether they are currently being converted to globally propagated ENQ requests. For more information, see “Reserve Monitor” in the zDMF installation manual.

10.4 Example migration in a shared DB2 test environment

In this example, active allocated data sets in a shared db2 environment are moved from 3390-3 to 3390-9. The following steps were performed:

- The storage group was prepared, source volumes were set to DISNEW, and new target volumes were set to ENABLE.
- Before target volumes were added to the storage group, table spaces were allocated at the source volumes and records include data were added.
- All records were checked.
- New target volumes added to the storage group and set to ENABLE, old source volumes defined as “DISNEW”.
- During migration updates were running against the table spaces out of two systems.
Groups for zDMF were created using BATCH.
Migration was monitored using the zDMF monitor.

After migration, all data were checked for consistency.

10.4.1 Test Environment Overview

Before starting, check that all prerequisites are met as described previously in this chapter. Figure 10-4 shows the System Managed Storage (SMS) volume status.

zDMF Test Scenario

zDMF was already installed and active as shown in Figure 10-5. For performance reasons, the zDMF communication data base and the user catalog for zDMF work data sets were on NONSMS managed volumes.

```
<table>
<thead>
<tr>
<th>Subsystem ZD</th>
<th>Host Messages</th>
<th>Row 1 to 2 of 2</th>
<th>Scroll === CSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMFID</td>
<td>SSID</td>
<td>CPUID</td>
<td>Last Server Cold Start Time</td>
</tr>
<tr>
<td>CVS2</td>
<td>ZDMF</td>
<td>05423A2064</td>
<td>06/24/2011 18:28:01.947</td>
</tr>
<tr>
<td>CEBC</td>
<td>ZDMF</td>
<td>04423A2064</td>
<td>06/24/2011 18:28:00.568</td>
</tr>
</tbody>
</table>
```

Figure 10-5 Check zDMF systems
These data sets can be checked by using the zDMF Monitor, as shown in Figure 10-6.

![Installation Options](image1)

You can also check the database and user catalog with TSO 3.4 with the High Level Qualifier (HLQ) of “CATALOG.ZDM*” (Figure 10-7).

![DSLIST - Data Sets Matching CATALOG.ZDM*](image2)
The alias for the work data sets used (HLQ of XXX) was added to the zDMF user catalog. Check for the volumes in storage group SG3390X1 and the SMS status as shown in Figure 10-8. Source volumes start with RS603x in the volume ID, and target volumes start with ZD850x.

```
SDSF ULOG CONSOLE STXXXXX1 LINE COMMAND ISSUED
COMMAND INPUT ===> /D SMS,SG(SG3390X1),LISTVOL SCROLL ===> CSR
RESPONSE=MCECEBC
IGD002I 00:34:31 DISPLAY SMS 172
STORGRP TYPE SYSTEM= 1 2
SG3390X1 POOL + +

<table>
<thead>
<tr>
<th>VOLUME</th>
<th>UNIT</th>
<th>SYSTEM= 1 2</th>
<th>STORGRP NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>RS603A</td>
<td>603A</td>
<td>D D</td>
<td>SG3390X1</td>
</tr>
<tr>
<td>RS603B</td>
<td>603B</td>
<td>D D</td>
<td>SG3390X1</td>
</tr>
<tr>
<td>RS603C</td>
<td>603C</td>
<td>D D</td>
<td>SG3390X1</td>
</tr>
<tr>
<td>RS603D</td>
<td>603D</td>
<td>D D</td>
<td>SG3390X1</td>
</tr>
<tr>
<td>RS603E</td>
<td>603E</td>
<td>D D</td>
<td>SG3390X1</td>
</tr>
<tr>
<td>RS603F</td>
<td>603F</td>
<td>D D</td>
<td>SG3390X1</td>
</tr>
<tr>
<td>RS603G</td>
<td>603G</td>
<td>D D</td>
<td>SG3390X1</td>
</tr>
<tr>
<td>RS603H</td>
<td>603H</td>
<td>D D</td>
<td>SG3390X1</td>
</tr>
<tr>
<td>RS603I</td>
<td>603I</td>
<td>D D</td>
<td>SG3390X1</td>
</tr>
<tr>
<td>RS603J</td>
<td>603J</td>
<td>D D</td>
<td>SG3390X1</td>
</tr>
<tr>
<td>RS603K</td>
<td>603K</td>
<td>D D</td>
<td>SG3390X1</td>
</tr>
<tr>
<td>RS603L</td>
<td>603L</td>
<td>D D</td>
<td>SG3390X1</td>
</tr>
<tr>
<td>RS603M</td>
<td>603M</td>
<td>D D</td>
<td>SG3390X1</td>
</tr>
<tr>
<td>RS603N</td>
<td>603N</td>
<td>D D</td>
<td>SG3390X1</td>
</tr>
<tr>
<td>RS603O</td>
<td>603O</td>
<td>D D</td>
<td>SG3390X1</td>
</tr>
<tr>
<td>RS603P</td>
<td>603P</td>
<td>D D</td>
<td>SG3390X1</td>
</tr>
<tr>
<td>RS603Q</td>
<td>603Q</td>
<td>D D</td>
<td>SG3390X1</td>
</tr>
<tr>
<td>ZD850A</td>
<td>850A</td>
<td>+ +</td>
<td>SG3390X1</td>
</tr>
<tr>
<td>ZD850B</td>
<td>850B</td>
<td>+ +</td>
<td>SG3390X1</td>
</tr>
<tr>
<td>ZD850C</td>
<td>850C</td>
<td>+ +</td>
<td>SG3390X1</td>
</tr>
<tr>
<td>ZD850D</td>
<td>850D</td>
<td>+ +</td>
<td>SG3390X1</td>
</tr>
<tr>
<td>ZD850E</td>
<td>850E</td>
<td>+ +</td>
<td>SG3390X1</td>
</tr>
<tr>
<td>ZD850F</td>
<td>850F</td>
<td>+ +</td>
<td>SG3390X1</td>
</tr>
</tbody>
</table>

*--------------------* LEGEND *--------------------*
. THE STORAGE GROUP OR VOLUME IS NOT DEFINED TO THE SYSTEM
+ THE STORAGE GROUP OR VOLUME IS ENABLED
- THE STORAGE GROUP OR VOLUME IS DISABLED
* THE STORAGE GROUP OR VOLUME IS QUIESCED
D THE STORAGE GROUP OR VOLUME IS DISABLED FOR NEW ALLOCATIONS ONLY
Q THE STORAGE GROUP OR VOLUME IS QUIESCED FOR NEW ALLOCATIONS ONLY
> THE VOLSER IN UCB IS DIFFERENT FROM THE VOLSER IN CONFIGURATION
SYSTEM  1 = MCECEBC    SYSTEM  2 = MZBCVS2
```

Figure 10-8  Displaying all volumes within storage group SG3390X1 with SMS status
Statistical data for the volume allocation status showed that all new volumes were empty (Figure 10-9).

<table>
<thead>
<tr>
<th>VOLUME</th>
<th>DEVI</th>
<th>TYPE</th>
<th>MOUNT</th>
<th>FREE SPACE</th>
<th>LARGEST FREE</th>
<th>PCT</th>
<th>FRAG</th>
</tr>
</thead>
<tbody>
<tr>
<td>RS603A</td>
<td>33903</td>
<td>603A</td>
<td>SMS X</td>
<td>1020</td>
<td>8</td>
<td>100%</td>
<td>30%</td>
</tr>
<tr>
<td>RS603B</td>
<td>33903</td>
<td>603B</td>
<td>SMS X</td>
<td>1462</td>
<td>1</td>
<td>14</td>
<td>100%</td>
</tr>
<tr>
<td>RS603C</td>
<td>33903</td>
<td>603C</td>
<td>SMS X</td>
<td>1616</td>
<td>11</td>
<td>1616</td>
<td>100%</td>
</tr>
<tr>
<td>RS603D</td>
<td>33903</td>
<td>603D</td>
<td>SMS X</td>
<td>568</td>
<td>7</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>RS603E</td>
<td>33903</td>
<td>603E</td>
<td>SMS X</td>
<td>1324</td>
<td>10</td>
<td>1324</td>
<td>100%</td>
</tr>
<tr>
<td>RS603F</td>
<td>33903</td>
<td>603F</td>
<td>SMS X</td>
<td>996</td>
<td>6</td>
<td>996</td>
<td>6</td>
</tr>
<tr>
<td>RS603G</td>
<td>33903</td>
<td>603G</td>
<td>SMS X</td>
<td>841</td>
<td>14</td>
<td>841</td>
<td>14</td>
</tr>
<tr>
<td>RS603H</td>
<td>33903</td>
<td>603H</td>
<td>SMS X</td>
<td>696</td>
<td>6</td>
<td>696</td>
<td>6</td>
</tr>
<tr>
<td>RS603I</td>
<td>33903</td>
<td>603I</td>
<td>SMS X</td>
<td>1764</td>
<td>10</td>
<td>1764</td>
<td>100%</td>
</tr>
<tr>
<td>RS603J</td>
<td>33903</td>
<td>603J</td>
<td>SMS X</td>
<td>1602</td>
<td>11</td>
<td>1602</td>
<td>100%</td>
</tr>
<tr>
<td>RS603K</td>
<td>33903</td>
<td>603K</td>
<td>SMS X</td>
<td>568</td>
<td>7</td>
<td>568</td>
<td>7</td>
</tr>
<tr>
<td>RS603L</td>
<td>33903</td>
<td>603L</td>
<td>SMS X</td>
<td>996</td>
<td>6</td>
<td>996</td>
<td>6</td>
</tr>
</tbody>
</table>

*Figure 10-9  Space Information for Storage Group “SG3390X1”*

**10.4.2 Preparing DSN List for Test Migration**

The test migration example started the migration group using BATCH JCL. A DSN list therefore needed to be created with data set names on the source volumes. The data sets were created using the following filter mask:

DSNCAT.DSNDBD.DBX1ST01.*
This filter mask results in 30 data sets, 15 tables, and 15 index data sets as shown in Figure 10-10.

<table>
<thead>
<tr>
<th>Command - Enter “/” to select action</th>
<th>Message</th>
<th>Volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNCAT.DSNDBD.DBX1ST01.IXA010A.I0001.A001</td>
<td></td>
<td>RS603A+</td>
</tr>
<tr>
<td>DSNCAT.DSNDBD.DBX1ST01.IXA010B.I0001.A001</td>
<td></td>
<td>RS6037+</td>
</tr>
<tr>
<td>DSNCAT.DSNDBD.DBX1ST01.IXA010C.I0001.A001</td>
<td></td>
<td>RS6031+</td>
</tr>
<tr>
<td>DSNCAT.DSNDBD.DBX1ST01.IXA010D.I0001.A001</td>
<td></td>
<td>RS603E+</td>
</tr>
<tr>
<td>DSNCAT.DSNDBD.DBX1ST01.IXA010E.I0001.A001</td>
<td></td>
<td>RS6035+</td>
</tr>
<tr>
<td>DSNCAT.DSNDBD.DBX1ST01.IXA010F.I0001.A001</td>
<td></td>
<td>RS6030+</td>
</tr>
<tr>
<td>DSNCAT.DSNDBD.DBX1ST01.IXA0101.I0001.A001</td>
<td></td>
<td>RS603C+</td>
</tr>
<tr>
<td>DSNCAT.DSNDBD.DBX1ST01.IXA0102.I0001.A001</td>
<td></td>
<td>RS6031+</td>
</tr>
<tr>
<td>DSNCAT.DSNDBD.DBX1ST01.IXA0103.I0001.A001</td>
<td></td>
<td>RS6030+</td>
</tr>
<tr>
<td>DSNCAT.DSNDBD.DBX1ST01.IXA0104.I0001.A001</td>
<td></td>
<td>RS6033+</td>
</tr>
<tr>
<td>DSNCAT.DSNDBD.DBX1ST01.IXA0105.I0001.A001</td>
<td></td>
<td>RS6038+</td>
</tr>
<tr>
<td>DSNCAT.DSNDBD.DBX1ST01.IXA0106.I0001.A001</td>
<td></td>
<td>RS6037+</td>
</tr>
<tr>
<td>DSNCAT.DSNDBD.DBX1ST01.IXA0107.I0001.A001</td>
<td></td>
<td>RS603F+</td>
</tr>
<tr>
<td>DSNCAT.DSNDBD.DBX1ST01.IXA0108.I0001.A001</td>
<td></td>
<td>RS603A+</td>
</tr>
<tr>
<td>DSNCAT.DSNDBD.DBX1ST01.IXA0109.I0001.A001</td>
<td></td>
<td>RS6033+</td>
</tr>
<tr>
<td>DSNCAT.DSNDBD.DBX1ST01.TSA010A.I0001.A001</td>
<td></td>
<td>RS603B+</td>
</tr>
<tr>
<td>DSNCAT.DSNDBD.DBX1ST01.TSA010B.I0001.A001</td>
<td></td>
<td>RS6030+</td>
</tr>
<tr>
<td>DSNCAT.DSNDBD.DBX1ST01.TSA010C.I0001.A001</td>
<td></td>
<td>RS6032+</td>
</tr>
<tr>
<td>DSNCAT.DSNDBD.DBX1ST01.TSA010D.I0001.A001</td>
<td></td>
<td>RS6034+</td>
</tr>
<tr>
<td>DSNCAT.DSNDBD.DBX1ST01.TSA010E.I0001.A001</td>
<td></td>
<td>RS6030+</td>
</tr>
<tr>
<td>DSNCAT.DSNDBD.DBX1ST01.TSA010F.I0001.A001</td>
<td></td>
<td>RS6032+</td>
</tr>
<tr>
<td>DSNCAT.DSNDBD.DBX1ST01.TSA0101.I0001.A001</td>
<td></td>
<td>RS6038+</td>
</tr>
<tr>
<td>DSNCAT.DSNDBD.DBX1ST01.TSA0102.I0001.A001</td>
<td></td>
<td>RS603F+</td>
</tr>
<tr>
<td>DSNCAT.DSNDBD.DBX1ST01.TSA0103.I0001.A001</td>
<td></td>
<td>RS6037+</td>
</tr>
<tr>
<td>DSNCAT.DSNDBD.DBX1ST01.TSA0104.I0001.A001</td>
<td></td>
<td>RS6035+</td>
</tr>
<tr>
<td>DSNCAT.DSNDBD.DBX1ST01.TSA0105.I0001.A001</td>
<td></td>
<td>RS6034+</td>
</tr>
<tr>
<td>DSNCAT.DSNDBD.DBX1ST01.TSA0106.I0001.A001</td>
<td></td>
<td>RS6031+</td>
</tr>
<tr>
<td>DSNCAT.DSNDBD.DBX1ST01.TSA0107.I0001.A001</td>
<td></td>
<td>RS6031+</td>
</tr>
<tr>
<td>DSNCAT.DSNDBD.DBX1ST01.TSA0108.I0001.A001</td>
<td></td>
<td>RS603A+</td>
</tr>
<tr>
<td>DSNCAT.DSNDBD.DBX1ST01.TSA0109.I0001.A001</td>
<td></td>
<td>RS6035+</td>
</tr>
</tbody>
</table>

Figure 10-10  Data set name list for test migration

10.4.3 Creating BATCH JCL

Figure 10-11 on page 464 shows a JCL example for zDMF batch. Split a long DSN list into groups of 16 data sets and start with a “SET” statement. It is helpful to use a REXX procedure to create this BATCH JCL out of an input DSN list.

The following are the zDMF parameters used in this test example:

- **DELETE_EXISTING_TARGET_DATASETS (NO):** Do not delete existing work data sets with same name.
- **EARLY_DATA_SET_COMPLETION (YES):** Divert and complete all non-allocated data sets as soon as MIRROR status is reached.
- **ALLOCSEQ (NONE):** Move data sets as they are in list rather than sort by size.
- **SPHERE (YES)**: Move all parts of a VSAM Cluster.
- **TOLERATE_SOURCE_VALIDATE_FAILURE (YES)** and **TOLERATE_TARGET_CREATE_FAILURE (YES)**: Do not stop the migration in case of an allocation error. Instead, skip this data set and proceed with next.

```
//STRZ999 JOB (A185,SYS),'STXXXXX IBM',
// MSGLEVEL=(1,1),NOTIFY=SYSUID,MSGCLASS=X,CLASS=A
//* IBM user name and telephone number
/*JOBPARM S=CEBC
//BATCH EXEC PGM=GZDBAT,PARM='CPFX=ZD'
//STEPLIB DD DISP=SHR,DSN=CPAC.HGZ0325.GZDLLIB
//SYSPRINT DD SYSOUT=* 
//SYSUDUMP DD SYSOUT=* 
//SYSIN DD *
IF GROUP TST999 EXISTS
COMMAND DELETE TST999
PREMOTE DLM=ZZ
GROUP (TST999)
- DELETE_EXISTING_TARGET_DATASETS (NO) -
- EARLY_DATA_SET_COMPLETION (YES) -
- ALOCSEQ (NONE)
SET -
SPHERE (YES) -
- TOLERATE_SOURCE_VALIDATE_FAILURE (YES) -
- SOURCE (DSN( 
- DSNCAT.DSNDBD.DBX1ST01.IXA01DA.I0001.A001 - 
- DSNCAT.DSNDBD.DBX1ST01.IXA010B.I0001.A001 - 
- DSNCAT.DSNDBD.DBX1ST01.IXA010C.I0001.A001 - 
- DSNCAT.DSNDBD.DBX1ST01.IXA010D.I0001.A001 - 
- DSNCAT.DSNDBD.DBX1ST01.IXA010E.I0001.A001 - 
- DSNCAT.DSNDBD.DBX1ST01.IXA010F.I0001.A001 - 
- DSNCAT.DSNDBD.DBX1ST01.IXA0101.I0001.A001 - 
- DSNCAT.DSNDBD.DBX1ST01.IXA0102.I0001.A001 - 
- DSNCAT.DSNDBD.DBX1ST01.IXA0103.I0001.A001 - 
- DSNCAT.DSNDBD.DBX1ST01.IXA0104.I0001.A001 - 
- DSNCAT.DSNDBD.DBX1ST01.IXA0105.I0001.A001 - 
- DSNCAT.DSNDBD.DBX1ST01.IXA0106.I0001.A001 - 
- DSNCAT.DSNDBD.DBX1ST01.IXA0107.I0001.A001 - 
- DSNCAT.DSNDBD.DBX1ST01.IXA0108.I0001.A001 - 
- DSNCAT.DSNDBD.DBX1ST01.IXA0109.I0001.A001 - 
- DSNCAT.DSNDBD.DBX1ST01.IXA010A.I0001.A001 )))-
- TOLERATE_TARGET_CREATE_FAILURE (YES) -
- TARGET (DSN (XXX))
SET -
SPHERE (YES) -
- TOLERATE_SOURCE_VALIDATE_FAILURE (YES) -
- SOURCE (DSN( 
- DSNCAT.DSNDBD.DBX1ST01.TSA010A.I0001.A001 - 
- DSNCAT.DSNDBD.DBX1ST01.TSA010B.I0001.A001 - 
- DSNCAT.DSNDBD.DBX1ST01.TSA010C.I0001.A001 - 
- DSNCAT.DSNDBD.DBX1ST01.TSA010D.I0001.A001 - 
- DSNCAT.DSNDBD.DBX1ST01.TSA010E.I0001.A001 - 
- DSNCAT.DSNDBD.DBX1ST01.TSA010F.I0001.A001 - 
- DSNCAT.DSNDBD.DBX1ST01.TSA0101.I0001.A001 - 
- DSNCAT.DSNDBD.DBX1ST01.TSA0102.I0001.A001 - 
- DSNCAT.DSNDBD.DBX1ST01.TSA0103.I0001.A001 - 
- DSNCAT.DSNDBD.DBX1ST01.TSA0104.I0001.A001 - 
- DSNCAT.DSNDBD.DBX1ST01.TSA0105.I0001.A001 - 
- DSNCAT.DSNDBD.DBX1ST01.TSA0106.I0001.A001 - 
- DSNCAT.DSNDBD.DBX1ST01.TSA0107.I0001.A001 - 
- DSNCAT.DSNDBD.DBX1ST01.TSA0108.I0001.A001 - 
- DSNCAT.DSNDBD.DBX1ST01.TSA0109.I0001.A001 )))-
- TOLERATE_TARGET_CREATE_FAILURE (YES) -
- TARGET (DSN (XXX))
ZZ
COMMAND ACTIVATE TST999
MAXWAIT 180
WAIT GROUP TST999 STATUS DEACT
IF RC = 0
QUIT
```

*Figure 10-11  zDMF Batch JCL Example*
10.4.4 Checking Data and Starting DB2 Load

To demonstrate a migration in an active system, Figure 10-12 shows all 15 table spaces (data consistency check). The record list includes system name, db2 system name, table space, table name, and a view of the first 10 changed records. The other records were checked as well. During an error, they are also displayed by the test procedure.

Each record was updated by two different systems (MCECEBC and MZBCVS2). The check was done from system MCECBE.

<table>
<thead>
<tr>
<th>REC</th>
<th>SYS</th>
<th>USERID</th>
<th>DATE</th>
<th>TIME</th>
<th>UPDATE</th>
<th>MARKED</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MCECEBC</td>
<td>STXXXXX1</td>
<td>20110710 19:53:27.706677</td>
<td>000052</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MZBCVS2</td>
<td>STXXXXX2</td>
<td>20110710 19:03:29.100314</td>
<td>000010</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>MCECEBC</td>
<td>STXXXXX1</td>
<td>20110710 19:53:27.706677</td>
<td>000052</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MZBCVS2</td>
<td>STXXXXX2</td>
<td>20110710 19:03:29.100314</td>
<td>000010</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>MCECEBC</td>
<td>STXXXXX1</td>
<td>20110710 19:53:27.706677</td>
<td>000052</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MZBCVS2</td>
<td>STXXXXX2</td>
<td>20110710 19:03:29.100314</td>
<td>000010</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>MCECEBC</td>
<td>STXXXXX1</td>
<td>20110710 19:53:27.706677</td>
<td>000052</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MZBCVS2</td>
<td>STXXXXX2</td>
<td>20110710 19:03:29.100314</td>
<td>000010</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>MCECEBC</td>
<td>STXXXXX1</td>
<td>20110710 19:53:27.706677</td>
<td>000052</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MZBCVS2</td>
<td>STXXXXX2</td>
<td>20110710 19:03:29.100314</td>
<td>000010</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>MCECEBC</td>
<td>STXXXXX1</td>
<td>20110710 19:53:27.706677</td>
<td>000052</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MZBCVS2</td>
<td>STXXXXX2</td>
<td>20110710 19:03:29.100314</td>
<td>000010</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>MCECEBC</td>
<td>STXXXXX1</td>
<td>20110710 19:53:27.706677</td>
<td>000052</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MZBCVS2</td>
<td>STXXXXX2</td>
<td>20110710 19:03:29.100314</td>
<td>000010</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>MCECEBC</td>
<td>STXXXXX1</td>
<td>20110710 19:53:27.706677</td>
<td>000052</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MZBCVS2</td>
<td>STXXXXX2</td>
<td>20110710 19:03:29.100314</td>
<td>000010</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>MCECEBC</td>
<td>STXXXXX1</td>
<td>20110710 19:53:27.706677</td>
<td>000052</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MZBCVS2</td>
<td>STXXXXX2</td>
<td>20110710 19:03:29.100314</td>
<td>000010</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>MCECEBC</td>
<td>STXXXXX1</td>
<td>20110710 19:53:27.706677</td>
<td>000052</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MZBCVS2</td>
<td>STXXXXX2</td>
<td>20110710 19:03:29.100314</td>
<td>000010</td>
<td>YES</td>
<td></td>
</tr>
</tbody>
</table>

Figure 10-12 Check data (window shot of last TS result, all others are the same)
Figure 10-13 shows updates from both z/OS systems to these data sets during the migration.

**Figure 10-13  SDSF activity display of db2 systems**

### 10.4.5 Starting and Monitoring Test Migration

Submit the BATCH JCL and monitor it using the zDMF Monitor item 1 + 2. The group is promoted by the BATCH job and automatically start activation. As soon activation is done, zDMF copies the data sets.

Figure 10-14 shows zDMF Monitor item 1.

**Figure 10-14  zDMF Monitor Item 1 (Manage Groups)**

The batch job will end after sending the data to the zDMF master started task.
The JCL log is shown in Figure 10-15.

```
.: IEF373I STEP/BATCH /START 2011193.0005
IEF032I STEP/BATCH /STOP 2011193.0009
  CPU: 0 HR 00 MIN 00.03 SEC SRB: 0 HR 00 MIN 00.00 SEC
IEF375I JOB/STRZ999 /START 2011193.0005
IEF033I JOB/STRZ999 /STOP 2011193.0009
  CPU: 0 HR 00 MIN 00.03 SEC SRB: 0 HR 00 MIN 00.00 SEC
DWRQBUF$ 2E104120 LN=00FFFE0 USED=00000000 REQ=00000000
==> IF GROUP ITS097 EXISTS
==> COMMAND DELETE ITS099
*** COMMAND SKIPPED DUE TO CONDITION TEST

==> PROMOTE DLM=ZZ
  GROUP (ITS099) -
    DELETE_EXISTING_TARGET_DATASETS (NO) -
    EARLY_DATA_SET_COMPLETION (YES) -
    ALLOCSEQ (NONE)
  SET -
    SPHERE (YES) -
    TOLERATE_SOURCE_VALIDATE_FAILURE (YES) -
    MIGRATE_ONLY_SPECIFIED_VOLUMES (NO) -
    SOURCE ( DSN( -
      DSNCAT.DSNDBD.DBX1ST01.IXA010A.I0001.A001 -
      DSNCAT.DSNDBD.DBX1ST01.IXA010B.I0001.A001 -
      DSNCAT.DSNDBD.DBX1ST01.TSA0108.I0001.A001 -
      DSNCAT.DSNDBD.DBX1ST01.TSA0109.I0001.A001 -
    )- -
  )
    TOLERATE_TARGET_CREATE_FAILURE (YES) -
    TARGET (DSN (XXX))
  :
*GROUP PROMOTED, STOW LENGTH IS 2256
*** RC=0

==> COMMAND ACTIVATE ITS099
  GROUP ITS099 ACTIVATED
*** RC=0

==> MAXWAIT 180
  MAXIMUM WAIT SET TO 180
*** RC=0

==> WAIT GROUP ITSO99 STATUS DEACT
  MAXIMUM WAIT EXCEEDED
*** RC=4

==> IF RC = 0

==> QUIT
*** COMMAND SKIPPED DUE TO CONDITION TEST
*** MAXIMUM RETURN CODE WAS 4

--> because of DELETE GROUP was skipt
```

Figure 10-15  zDMF Batch JCL output example
After group is promoted, zDMF checks each data set in the list. It fetches the actual detailed data set information, number of volumes, multi-volume DS, size, number of extends, and so on. zDMF also discovers the available space at the new target volumes for each single data set. After the data sets are validated, the status is changed to Active as shown in Figure 10-16.

As soon all data sets (source and work DS) are Active, zDMF changes to Pending-Active (Figure 10-17).

The Copy phase starts automatically (Figure 10-18).
When the data is migrated, the status changes to **Moved** as shown in Figure 10-19.

<table>
<thead>
<tr>
<th>Subsystem ZD</th>
<th>Groups</th>
<th>Row 1 to 3 of 3</th>
<th>Scroll ====&gt; CSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Set</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Extents</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>_ ITS099</td>
<td>Owning System: CEBC</td>
<td>Moved=100% Mirror</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Activate report available</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Early Data Set Complete status: 000016 Data sets diverted.**

*Figure 10-19 zDMF Monitor Item 2 (Interact with Promoted Groups): Mirror*
Data Migration to IBM Disk Storage Systems

After Mirror status is reached, a sequence starts to move non-allocated data sets (ESDC=YES). Figure 10-20 shows that 16 data sets are already moved from the source data sets.

**Figure 10-20  Data set status (source data sets)**
Similarly, Figure 10-21 shows the data sets on the work side. Because the VSAM data sets are moved, the VSAM CLUSTER name is not displayed. You will find data parts only for source and work data sets. In the example, eight data sets are already moved, and the VSAM Cluster has no relation to a volume.

<table>
<thead>
<tr>
<th>Command - Enter &quot;/&quot; to select action</th>
<th>Message</th>
<th>Volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>XXX.ITS099.D1193.T0020546.S00001.DATA</td>
<td>ZD850F+</td>
<td></td>
</tr>
<tr>
<td>XXX.ITS099.D1193.T0020546.S00002.DATA</td>
<td>ZD850B+</td>
<td></td>
</tr>
<tr>
<td>XXX.ITS099.D1193.T0020546.S00003.DATA</td>
<td>ZD850C+</td>
<td></td>
</tr>
<tr>
<td>XXX.ITS099.D1193.T0020546.S00004.DATA</td>
<td>ZD850C+</td>
<td></td>
</tr>
<tr>
<td>XXX.ITS099.D1193.T0020546.S00005.DATA</td>
<td>ZD850D+</td>
<td></td>
</tr>
<tr>
<td>XXX.ITS099.D1193.T0020546.S00006.DATA</td>
<td>ZD850C+</td>
<td></td>
</tr>
<tr>
<td>XXX.ITS099.D1193.T0020546.S00007.DATA</td>
<td>ZD850F+</td>
<td></td>
</tr>
<tr>
<td>XXX.ITS099.D1193.T0020546.S00008.DATA</td>
<td>ZD850F+</td>
<td></td>
</tr>
<tr>
<td>XXX.ITS099.D1193.T0020546.S00009.DATA</td>
<td>ZD850E+</td>
<td></td>
</tr>
<tr>
<td>XXX.ITS099.D1193.T0020546.S00010.DATA</td>
<td>ZD850B+</td>
<td></td>
</tr>
<tr>
<td>XXX.ITS099.D1193.T0020546.S00011.DATA</td>
<td>ZD850C+</td>
<td></td>
</tr>
<tr>
<td>XXX.ITS099.D1193.T0020546.S00012.DATA</td>
<td>ZD850F+</td>
<td></td>
</tr>
<tr>
<td>XXX.ITS099.D1193.T0020546.S00013.DATA</td>
<td>ZD850F+</td>
<td></td>
</tr>
<tr>
<td>XXX.ITS099.D1193.T0020546.S00014.DATA</td>
<td>ZD850D+</td>
<td></td>
</tr>
<tr>
<td>XXX.ITS099.D1193.T0020546.S00015.DATA</td>
<td>ZD850B+</td>
<td></td>
</tr>
<tr>
<td>XXX.ITS099.D1193.T0020546.S00016.DATA</td>
<td>ZD850C+</td>
<td></td>
</tr>
<tr>
<td>XXX.ITS099.D1193.T0020546.S00017.DATA</td>
<td>ZD850C+</td>
<td></td>
</tr>
<tr>
<td>XXX.ITS099.D1193.T0020546.S00018.DATA</td>
<td>ZD850E+</td>
<td></td>
</tr>
<tr>
<td>XXX.ITS099.D1193.T0020546.S00019.DATA</td>
<td>ZD850F+</td>
<td></td>
</tr>
<tr>
<td>XXX.ITS099.D1193.T0020546.S00020.DATA</td>
<td>ZD850D+</td>
<td></td>
</tr>
<tr>
<td>XXX.ITS099.D1193.T0020546.S00021.DATA</td>
<td>ZD850C+</td>
<td></td>
</tr>
<tr>
<td>XXX.ITS099.D1193.T0020546.S00022.DATA</td>
<td>RS6038+</td>
<td></td>
</tr>
<tr>
<td>XXX.ITS099.D1193.T0020546.S00023.DATA</td>
<td>RS6038+</td>
<td></td>
</tr>
<tr>
<td>XXX.ITS099.D1193.T0020546.S00024.DATA</td>
<td>RS6037+</td>
<td></td>
</tr>
<tr>
<td>XXX.ITS099.D1193.T0020546.S00025.DATA</td>
<td>RS6035+</td>
<td></td>
</tr>
<tr>
<td>XXX.ITS099.D1193.T0020546.S00026.DATA</td>
<td>RS6034+</td>
<td></td>
</tr>
<tr>
<td>XXX.ITS099.D1193.T0020546.S00027.DATA</td>
<td>RS6031+</td>
<td></td>
</tr>
<tr>
<td>XXX.ITS099.D1193.T0020546.S00028.DATA</td>
<td>RS6031+</td>
<td></td>
</tr>
<tr>
<td>XXX.ITS099.D1193.T0020546.S00029.DATA</td>
<td>RS603A+</td>
<td></td>
</tr>
<tr>
<td>XXX.ITS099.D1193.T0020546.S00030.DATA</td>
<td>ZD850C+</td>
<td></td>
</tr>
<tr>
<td>XXX.ITS099.D1193.T0020546.S00031.DATA</td>
<td>ZD850E+</td>
<td></td>
</tr>
</tbody>
</table>

Figure 10-21  Data set status (work data sets)
10.4.6 Diverting the Group

Started the divert process by entering V for divert in front of the group as shown in Figure 10-22.

```
<table>
<thead>
<tr>
<th>Subsystem ZD</th>
<th>Groups</th>
<th>Row 1 to 3 of 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group</td>
<td>CEBC</td>
<td>Status</td>
</tr>
<tr>
<td>Data Set</td>
<td></td>
<td>Extents</td>
</tr>
<tr>
<td>v ITSO99</td>
<td>Owning System: CEBC</td>
<td>Moved=100% Mirror</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Activate report available</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Early Data Set Complete status: 000016 Data sets diverted.</td>
</tr>
</tbody>
</table>

```

```
<table>
<thead>
<tr>
<th>Subsystem ZD</th>
<th>Groups</th>
<th>Row 1 to 2 of 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group</td>
<td>CEBC</td>
<td>Status</td>
</tr>
<tr>
<td>Data Set</td>
<td></td>
<td>Extents</td>
</tr>
<tr>
<td>– ITSO99</td>
<td>Owning System: CEBC</td>
<td>Diverted= 25% P-Divert</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Activate report available</td>
</tr>
</tbody>
</table>
```

Figure 10-22  ZDMF Monitor Item 2 - start divert

To see more details by modifying the display options, press the F4 key and modify the first three lines (Figure 10-23).

```
<table>
<thead>
<tr>
<th>Display Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command ==&gt;</td>
</tr>
<tr>
<td>Show Source Data Set Name . . . . Y or N . . y  &lt;--</td>
</tr>
<tr>
<td>Show Target Data Set Name . . . . Y or N . . y  &lt;--</td>
</tr>
<tr>
<td>Show Extent Information . . . . Y or N . . y     &lt;--</td>
</tr>
<tr>
<td>Show Command Messages . . . . . . Y or N . . N</td>
</tr>
<tr>
<td>Show Command Diagnostic Information . Y or N . . N</td>
</tr>
</tbody>
</table>

F1=Help  F5=Save Settings  F3=Exit
```

Figure 10-23  ZDMF Monitor Item 2 (modify display options)

Save your new settings by pressing F4.
Pressing F3 shows the new details as shown in Figure 10-24. The details include the number of all extends for each data set (source and target), and cylinder and head information. VSAM Cluster information is also displayed.

```
<table>
<thead>
<tr>
<th>Subsystem ZD</th>
<th>Groups</th>
<th>Row 1 to 35 of 372</th>
<th>Command ==&gt;</th>
<th>Scroll ==&gt;</th>
<th>CSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group</td>
<td></td>
<td></td>
<td>CEBC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Set</td>
<td></td>
<td></td>
<td>Status</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Extents</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>ITS099</em></td>
<td>Owning System: CEBC</td>
<td>n/a</td>
<td>Divert</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Activate and Divert reports available</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Src:DSNCAT.DSNDBC.DBX1ST01.IXA010A.I0001.A001</em></td>
<td>VSAM Cluster</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tar:XXX.ITS099.D1193.T0020546.S00001</td>
<td>VSAM Cluster</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Src:DSNCAT.DSNDBC.DBX1ST01.IXA010B.I0001.A001</em></td>
<td>VSAM Cluster</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tar:XXX.ITS099.D1193.T0020546.S00002</td>
<td>VSAM Cluster</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Src:DSNCAT.DSNDBD.DBX1ST01.TSA010F.I0001.A001</em></td>
<td>VSAM Cluster</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unit=6032(RS6032) Cyl=000004D Hd=B</td>
<td>Divert</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unit=6032(RS6032) Cyl=0000209 Hd=5</td>
<td>Divert</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unit=6032(RS6032) Cyl=0000307 Hd=5</td>
<td>Divert</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unit=6032(RS6032) Cyl=000040FD Hd=5</td>
<td>Divert</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tar:XXX.ITS099.D1193.T0020546.S00021.DATA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unit=850C(ZD850C) Cyl=000015B Hd=1 #Trks=00000976 Moved=100%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unit=850C(ZD850C) Cyl=000019C Hd=2 #Trks=00000375 Moved=100%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unit=850C(ZD850C) Cyl=00001b5 Hd=2 #Trks=00000435 Moved=100%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unit=850C(ZD850C) Cyl=00001d2 Hd=2 #Trks=00000495 Moved=100%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Src:DSNCAT.DSNDBD.DBX1ST01.TSA0101.I0001.A001</em></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unit=6038(RS6038) Cyl=000000C Hd=0</td>
<td>Complete</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unit=6038(RS6038) Cyl=000000C Hd=B</td>
<td>Complete</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unit=6038(RS6038) Cyl=0000308A Hd=0</td>
<td>Complete</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unit=6038(RS6038) Cyl=000083B Hd=0</td>
<td>Complete</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unit=6038(RS6038) Cyl=000083A Hd=0</td>
<td>Complete</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tar:XXX.ITS099.D1193.T0020546.S00022.DATA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unit=850C(ZD850C) Cyl=000001F3 Hd=2 #Trks=00000001 Moved=100%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unit=850C(ZD850C) Cyl=00001f3 Hd=3 #Trks=00000975 Moved=100%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unit=850C(ZD850C) Cyl=0000234 Hd=3 #Trks=00000375 Moved=100%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unit=850C(ZD850C) Cyl=000024D Hd=3 #Trks=00000435 Moved=100%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unit=850C(ZD850C) Cyl=000026A Hd=3 #Trks=00000495 Moved=100%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Src:DSNCAT.DSNDBD.DBX1ST01.TSA0102.I0001.A001</em></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unit=603f(RS603f) Cyl=000000C Hd=0</td>
<td>Complete</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unit=603f(RS603f) Cyl=0000017 Hd=0</td>
<td>Complete</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unit=603f(RS603f) Cyl=000040D Hd=9</td>
<td>Complete</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unit=603f(RS603f) Cyl=0000584 Hd=9</td>
<td>Complete</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unit=603f(RS603f) Cyl=0000737 Hd=9</td>
<td>Complete</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tar:XXX.ITS099.D1193.T0020546.S00023.DATA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unit=850C(ZD850C) Cyl=000015B Hd=1 #Trks=00000976 Moved=100%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unit=850C(ZD850C) Cyl=000019C Hd=2 #Trks=00000375 Moved=100%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unit=850C(ZD850C) Cyl=00001b5 Hd=2 #Trks=00000435 Moved=100%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unit=850C(ZD850C) Cyl=00001d2 Hd=2 #Trks=00000495 Moved=100%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 10-24   zDMF Monitor Detailed extend list of an active group
To see which data sets are allocated by application, enter J in front of the group name. The details on the system SMFID and the application names allocating the data sets are displayed (Figure 10-25).

<table>
<thead>
<tr>
<th>Command</th>
<th>Data Set Name</th>
<th>SMFID</th>
<th>Row 1 to 37 of 37</th>
<th>Scroll</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSR</td>
<td>DSNCAT.DSNDBC.DBX1ST01.IXA010A.I0001.A001</td>
<td>CVS2</td>
<td>DB2ADBM1</td>
<td>CSR</td>
</tr>
<tr>
<td></td>
<td>DSNCAT.DSNDBC.DBX1ST01.IXA010B.I0001.A001</td>
<td>CVS2</td>
<td>DB2ADBM1</td>
<td>CSR</td>
</tr>
<tr>
<td></td>
<td>DSNCAT.DSNDBC.DBX1ST01.IXA010C.I0001.A001</td>
<td>CVS2</td>
<td>DB2ADBM1</td>
<td>CSR</td>
</tr>
<tr>
<td></td>
<td>DSNCAT.DSNDBC.DBX1ST01.IXA010D.I0001.A001</td>
<td>CVS2</td>
<td>DB2ADBM1</td>
<td>CSR</td>
</tr>
<tr>
<td></td>
<td>DSNCAT.DSNDBC.DBX1ST01.IXA010E.I0001.A001</td>
<td>CVS2</td>
<td>DB2ADBM1</td>
<td>CSR</td>
</tr>
<tr>
<td></td>
<td>DSNCAT.DSNDBC.DBX1ST01.IXA010F.I0001.A001</td>
<td>CVS2</td>
<td>DB2ADBM1</td>
<td>CSR</td>
</tr>
<tr>
<td></td>
<td>DSNCAT.DSNDBC.DBX1ST01.IXA0101.I0001.A001</td>
<td>CVS2</td>
<td>DB2ADBM1</td>
<td>CSR</td>
</tr>
<tr>
<td></td>
<td>DSNCAT.DSNDBC.DBX1ST01.IXA0102.I0001.A001</td>
<td>CVS2</td>
<td>DB2ADBM1</td>
<td>CSR</td>
</tr>
<tr>
<td></td>
<td>DSNCAT.DSNDBC.DBX1ST01.IXA0103.I0001.A001</td>
<td>CVS2</td>
<td>DB2ADBM1</td>
<td>CSR</td>
</tr>
<tr>
<td></td>
<td>DSNCAT.DSNDBC.DBX1ST01.IXA0104.I0001.A001</td>
<td>CVS2</td>
<td>DB2ADBM1</td>
<td>CSR</td>
</tr>
<tr>
<td></td>
<td>DSNCAT.DSNDBC.DBX1ST01.IXA0105.I0001.A001</td>
<td>CVS2</td>
<td>DB2ADBM1</td>
<td>CSR</td>
</tr>
<tr>
<td></td>
<td>DSNCAT.DSNDBC.DBX1ST01.IXA0106.I0001.A001</td>
<td>CVS2</td>
<td>DB2ADBM1</td>
<td>CSR</td>
</tr>
<tr>
<td></td>
<td>DSNCAT.DSNDBC.DBX1ST01.IXA0107.I0001.A001</td>
<td>CVS2</td>
<td>DB2ADBM1</td>
<td>CSR</td>
</tr>
<tr>
<td></td>
<td>DSNCAT.DSNDBC.DBX1ST01.IXA0108.I0001.A001</td>
<td>CVS2</td>
<td>DB2ADBM1</td>
<td>CSR</td>
</tr>
<tr>
<td></td>
<td>DSNCAT.DSNDBC.DBX1ST01.IXA0109.I0001.A001</td>
<td>CVS2</td>
<td>DB2ADBM1</td>
<td>CSR</td>
</tr>
<tr>
<td></td>
<td>DSNCAT.DSNDBC.DBX1ST01.TSA010A.I0001.A001</td>
<td>CVS2</td>
<td>DB2ADBM1</td>
<td>CSR</td>
</tr>
<tr>
<td></td>
<td>DSNCAT.DSNDBC.DBX1ST01.TSA010B.I0001.A001</td>
<td>CVS2</td>
<td>DB2ADBM1</td>
<td>CSR</td>
</tr>
<tr>
<td></td>
<td>DSNCAT.DSNDBC.DBX1ST01.TSA010C.I0001.A001</td>
<td>CVS2</td>
<td>DB2ADBM1</td>
<td>CSR</td>
</tr>
<tr>
<td></td>
<td>DSNCAT.DSNDBC.DBX1ST01.TSA010D.I0001.A001</td>
<td>CVS2</td>
<td>DB2ADBM1</td>
<td>CSR</td>
</tr>
<tr>
<td></td>
<td>DSNCAT.DSNDBC.DBX1ST01.TSA010E.I0001.A001</td>
<td>CVS2</td>
<td>DB2ADBM1</td>
<td>CSR</td>
</tr>
<tr>
<td></td>
<td>DSNCAT.DSNDBC.DBX1ST01.TSA010F.I0001.A001</td>
<td>CVS2</td>
<td>DB2ADBM1</td>
<td>CSR</td>
</tr>
<tr>
<td></td>
<td>DSNCAT.DSNDBC.DBX1ST02.IXA020C.I0001.A001</td>
<td>CVS2</td>
<td>DB2ADBM1</td>
<td>CSR</td>
</tr>
</tbody>
</table>

*Figure 10-25 zDMF detailed data sets allocation list (zDMF monitor)*

Press F3 to go back to the normal monitor display.
In a production move, stay with this status until all data sets in a storage group are moved. In the test environment, db2 updates were run to other table spaces overnight. The allocation status changed to the status seen in Figure 10-26.

In cooperation with the DB2 team, you can complete this group in different ways. You can close the single table spaces, or you can also stop one DB2 after the other. Wait until all data sets are in divert status or already moved, and all metadata (Catalog, VTOC and VVDS) are updated. You can restart of DB2 immediately.

You need to schedule a time, usually on the weekend or during the night, to stop application for a short time. This outage is required to release allocation to the data sets. As long one single data set of one session is allocated, the session does not end. As soon application is accessing the catalog migration of all data sets in status DIVERT, the migration is complete. Check with zDMF monitor that the group has no data sets allocated as shown in Figure 10-27.

---

### Figure 10-26  zDMF detailed data sets allocation list (zDMF monitor) next day

<table>
<thead>
<tr>
<th>Data Set Name</th>
<th>SMFID</th>
<th>Job Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNCAT.DSNDBC.DBX1ST01.IXA01OA.I0001.A001</td>
<td>CVS2</td>
<td>DB2ADBM1</td>
</tr>
<tr>
<td>DSNCAT.DSNDBC.DBX1ST01.IXA01OB.I0001.A001</td>
<td>CVS2</td>
<td>DB2ADBM1</td>
</tr>
<tr>
<td>DSNCAT.DSNDBC.DBX1ST01.IXA01OC.I0001.A001</td>
<td>CVS2</td>
<td>DB2ADBM1</td>
</tr>
<tr>
<td>DSNCAT.DSNDBC.DBX1ST01.IXA01OD.I0001.A001</td>
<td>CVS2</td>
<td>DB2ADBM1</td>
</tr>
<tr>
<td>DSNCAT.DSNDBC.DBX1ST01.IXA01OE.I0001.A001</td>
<td>CVS2</td>
<td>DB2ADBM1</td>
</tr>
<tr>
<td>DSNCAT.DSNDBC.DBX1ST01.IXA01OF.I0001.A001</td>
<td>CVS2</td>
<td>DB2ADBM1</td>
</tr>
<tr>
<td>DSNCAT.DSNDBC.DBX1ST01.IXA0101.I0001.A001</td>
<td>CVS2</td>
<td>DB2ADBM1</td>
</tr>
<tr>
<td>DSNCAT.DSNDBC.DBX1ST01.IXA0102.I0001.A001</td>
<td>CVS2</td>
<td>DB2ADBM1</td>
</tr>
<tr>
<td>DSNCAT.DSNDBC.DBX1ST01.IXA0103.I0001.A001</td>
<td>CVS2</td>
<td>DB2ADBM1</td>
</tr>
<tr>
<td>DSNCAT.DSNDBC.DBX1ST01.IXA0104.I0001.A001</td>
<td>CVS2</td>
<td>DB2ADBM1</td>
</tr>
<tr>
<td>DSNCAT.DSNDBC.DBX1ST01.IXA0105.I0001.A001</td>
<td>CVS2</td>
<td>DB2ADBM1</td>
</tr>
<tr>
<td>DSNCAT.DSNDBC.DBX1ST01.IXA0106.I0001.A001</td>
<td>CVS2</td>
<td>DB2ADBM1</td>
</tr>
<tr>
<td>DSNCAT.DSNDBC.DBX1ST01.IXA0107.I0001.A001</td>
<td>CVS2</td>
<td>DB2ADBM1</td>
</tr>
<tr>
<td>DSNCAT.DSNDBC.DBX1ST01.IXA0108.I0001.A001</td>
<td>CVS2</td>
<td>DB2ADBM1</td>
</tr>
<tr>
<td>DSNCAT.DSNDBC.DBX1ST01.IXA0109.I0001.A001</td>
<td>CVS2</td>
<td>DB2ADBM1</td>
</tr>
<tr>
<td>DSNCAT.DSNDBC.DBX1ST01.TSA010A.I0001.A001</td>
<td>CEBC</td>
<td>DB1ADBM1</td>
</tr>
<tr>
<td>DSNCAT.DSNDBC.DBX1ST01.TSA010B.I0001.A001</td>
<td>CEBC</td>
<td>DB1ADBM1</td>
</tr>
<tr>
<td>DSNCAT.DSNDBC.DBX1ST01.TSA010C.I0001.A001</td>
<td>CEBC</td>
<td>DB1ADBM1</td>
</tr>
<tr>
<td>DSNCAT.DSNDBC.DBX1ST01.TSA010D.I0001.A001</td>
<td>CEBC</td>
<td>DB1ADBM1</td>
</tr>
<tr>
<td>DSNCAT.DSNDBC.DBX1ST01.TSA010E.I0001.A001</td>
<td>CEBC</td>
<td>DB1ADBM1</td>
</tr>
<tr>
<td>DSNCAT.DSNDBC.DBX1ST01.TSA010F.I0001.A001</td>
<td>CEBC</td>
<td>DB1ADBM1</td>
</tr>
</tbody>
</table>

---

### Figure 10-27  zDMF Monitor data set allocation status after application was stopped

<table>
<thead>
<tr>
<th>Data Set Name</th>
<th>SMFID</th>
<th>Job Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>No jobs allocated to any data sets included in this group.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
In addition, make sure that the process is completed as shown in Figure 10-28.

![Figure 10-28  zDMF Monitor: Status of group is Complete](image)

After group reaches the status Complete, check the new location of the moved and work data sets as shown in Figure 10-29.

![Figure 10-29  Location of source and work data sets](image)
Also, test the data consistency by running the test procedure with the check parameter. All tables were moved to new location without any problem as shown in Figure 10-30.

| -- SYSTEM          | MCECEBC         |
| -- DB2 SYSTEM NAME | DB1A            |
| -- DB2 TABLE SPACE | TSA010F         |
| -- DB2 TABLE       | TA010F          |
| -- NUMBER OF RECORDS IN TABLE | 25000 |
| -- CHECK ALL RECORDS WHERE FIELD CHANGED = YES |
| -- DISPLAY NUMBER OF UPDATED RECORDS (FIRST 10) |

| REC-----SYS------USERID---DATE-----TIME------------UPDATE-MARKED-- |
|----------MCECEBC  STXXXXX1 20110712 18:13:55.728986 000064 YES |
| ------ MZBCVS2  STXXXXX2 20110712 18:56:06.467727 000050 YES |
| 000010 MCECEBC  STXXXXX1 20110712 18:13:55.728986 000064 YES |
| ------ MZBCVS2  STXXXXX2 20110712 18:56:06.467727 000050 YES |
| 000013 MCECEBC  STXXXXX1 20110712 18:13:55.728986 000064 YES |
| ------ MZBCVS2  STXXXXX2 20110712 18:56:06.467727 000050 YES |
| 000025 MCECEBC  STXXXXX1 20110712 18:13:55.728986 000064 YES |
| ------ MZBCVS2  STXXXXX2 20110712 18:56:06.467727 000050 YES |
| 000039 MCECEBC  STXXXXX1 20110712 18:13:55.728986 000064 YES |
| ------ MZBCVS2  STXXXXX2 20110712 18:56:06.467727 000050 YES |
| 000065 MCECEBC  STXXXXX1 20110712 18:13:55.728986 000064 YES |
| ------ MZBCVS2  STXXXXX2 20110712 18:56:06.467727 000050 YES |
| 000070 MCECEBC  STXXXXX1 20110712 18:13:55.728986 000064 YES |
| ------ MZBCVS2  STXXXXX2 20110712 18:56:06.467727 000050 YES |
| 000076 MCECEBC  STXXXXX1 20110712 18:13:55.728986 000064 YES |
| ------ MZBCVS2  STXXXXX2 20110712 18:56:06.467727 000050 YES |
| 000093 MCECEBC  STXXXXX1 20110712 18:13:55.728986 000064 YES |
| ------ MZBCVS2  STXXXXX2 20110712 18:56:06.467727 000050 YES |
| 000105 MCECEBC  STXXXXX1 20110712 18:13:55.728986 000064 YES |
| ------ MZBCVS2  STXXXXX2 20110712 18:56:06.467727 000050 YES |

--- TOTAL RECORDS -----------------------------------------------
-- MARKED RECORDS = 2500
-- OTHER RECORDS = 0
-- MARKED RECORDS = ALL RECORDS OK
-- OTHER RECORDS = NO PROBLEMS FOUND

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

Figure 10-30  Test procedure, example result for the last table after migration completed
10.4.7 Starting post migration cleanup procedure

To perform post migration, enter Z in front of the group name as shown in Figure 10-31.

<table>
<thead>
<tr>
<th>Subsystem</th>
<th>ZD</th>
<th>Groups</th>
<th>Row 2 to 3 of 3</th>
<th>Scroll ====&gt;</th>
<th>CSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command</td>
<td>====&gt;</td>
<td>Group</td>
<td>CEBE</td>
<td>Data Set</td>
<td>Extents</td>
</tr>
<tr>
<td></td>
<td></td>
<td>z ITSO99</td>
<td>Activate, Divert, and Terminate reports available</td>
<td>Complete</td>
<td></td>
</tr>
</tbody>
</table>

Figure 10-31  Start Cleanup procedure with line command “Z”

Based on the global setting of the zDMF installation, you can delete the work data sets in one of two ways:

- Change the setting for Create ICKDSF TRKFMT Statements to Y as shown in Figure 10-32.

<table>
<thead>
<tr>
<th>User Session Options</th>
<th>07/17/11 01:25:09</th>
</tr>
</thead>
<tbody>
<tr>
<td>zDMF Command Prefix</td>
<td>ZD</td>
</tr>
<tr>
<td>Group Definition Data Set</td>
<td>STXXXXX.ZDMF.CONFIG</td>
</tr>
<tr>
<td>Use Log Data Set</td>
<td>N (Y/N) Retain 07 generations of logs (01-30)</td>
</tr>
<tr>
<td>Use Browse or View</td>
<td>V (B/V)</td>
</tr>
<tr>
<td>Log Data Set Name</td>
<td>__________________</td>
</tr>
<tr>
<td>Messages with Local or GMT time</td>
<td>L (L/G)</td>
</tr>
<tr>
<td>Show Command Messages</td>
<td>N (Y/N)</td>
</tr>
<tr>
<td>Show Command Diagnostic Info</td>
<td>N (Y/N)</td>
</tr>
<tr>
<td><strong>Create ICKDSF TRKFMT Statements.</strong> Y (Y/N) -- “Y”</td>
<td></td>
</tr>
<tr>
<td>Early Data Set Completion</td>
<td>Y (Y/N)</td>
</tr>
<tr>
<td>zDMF Load Library</td>
<td>CPAC.HGZD325.GZDLLIB</td>
</tr>
<tr>
<td>zDMF Rexx Library</td>
<td>CPAC.HGZD325.GZDELIB</td>
</tr>
<tr>
<td>zDMF Panel Library</td>
<td>CPAC.HGZD325.GZDPLIB</td>
</tr>
<tr>
<td>zDMF Table Library</td>
<td>CPAC.HGZD325.GZDTLIB</td>
</tr>
<tr>
<td>zDMF Message Library</td>
<td>CPAC.HGZD325.GZDMLIB</td>
</tr>
<tr>
<td>zDMF Security Library</td>
<td>CPAC.HGZD325.GZDLLIB</td>
</tr>
</tbody>
</table>

Figure 10-32  zDMF Global setting (zDMF monitor time 6) for ICKDSF TRKFMT

The group TDMF creates a job with ICKDSF track format statements that overwrite all work data sets with dummy data for each data set migrated. The work data sets and extend maps created by zDMF are deleted as shown in Figure 10-33 on page 479. You must update the job card information to your requirements and submit this job.

**Remember:** This ICKDSF job starts heavy I/O load to the volumes containing the work data sets.
Figure 10-33  Example JCL to start ICKDSF (overwrite work data sets)
Change the setting for Create ICKDSF TRKFMT Statements to N as shown in Figure 10-34 to create an IDCAMS job with delete statements.

**Important:** Do not delete Extend Maps when zDMF is still active with running groups. Delete this part of the job, work data sets of completed groups can be deleted.

```
Figure 10-34   zDMF Global setting (zDMF monitor time 6) for IDCAMS delete statements
```

User Session Options

<table>
<thead>
<tr>
<th>Command</th>
<th>07/17/11 01:25:09</th>
</tr>
</thead>
<tbody>
<tr>
<td>zDMF Command Prefix</td>
<td>ZD</td>
</tr>
<tr>
<td>Group Definition Data Set</td>
<td>STXXXXX.ZDMF.CONFIG</td>
</tr>
<tr>
<td>Use Log Data Set</td>
<td>N (Y/N) Retain 07 generations of logs (01-30)</td>
</tr>
<tr>
<td>Use Browse or View</td>
<td>V (B/V)</td>
</tr>
<tr>
<td>Log Data Set Name</td>
<td></td>
</tr>
<tr>
<td>Messages with Local or GMT time</td>
<td>L (L/G)</td>
</tr>
<tr>
<td>Show Command Messages</td>
<td>N (Y/N)</td>
</tr>
<tr>
<td>Show Command Diagnostic Info</td>
<td>N (Y/N)</td>
</tr>
<tr>
<td>Create ICKDSF TRKFMT Statements</td>
<td>N (Y/N) &lt;-- &quot;N&quot;</td>
</tr>
<tr>
<td>Early Data Set Completion</td>
<td>Y (Y/N)</td>
</tr>
</tbody>
</table>

zDMF Load Library      CPAC.HGZD325.GZDLLIB
zDMF Rexx Library      CPAC.HGZD325.GZDELIB
zDMF Panel Library     CPAC.HGZD325.GZDPLIB
zDMF Table Library     CPAC.HGZD325.GZDTLIB
zDMF Message Library   CPAC.HGZD325.GZDMLIB
zDMF Security Library  CPAC.HGZD325.GZDLLIB

F1=Help     F3=Exit     F5=Save Settings     F12 = Cancel
Enter Z next to the group to begin the deletion process as shown in Figure 10-35.

```
VIEW       SYS11198.T013954.RA000.STXXXXX.R0147421     Columns 00001 00072
Command ===>                                                  Scroll ===> PAGE
****** **************************** Top of Data ****************************
000001 Jobcard1        Have sufficient region size
000002 Jobcard2        to execute IDCAMS
000003 /*
000004 /*  You may submit the jcl from this panel after you
000005 /*  update the job card(s). To save a copy, you must use
000006 /*  the CREATE or REPLACE primary command to place the
000007 /*  data in a data set of your choosing.
000008 /*  Reference the ISPF User's Guide on the use of VIEW.
000009 /*
000010 /*  All statements should be reviewed prior to submitting.
000011 /*
000012 /*
000013 /*  User MUST have SAF authority to delete these data sets.
000014 /*
000015 //DELETES  EXEC PGM=IDCAMS
000016 //SYSPRINT DD SYSOUT=*                
000017 //SYSIN     DD *
000018
000019 /* GROUP NAME: ITSO99
            DELETE STATEMENTS FOR DATA SETS SUCCESSFULLY MIGRATED. /*
000020
000021
000022 DELETE 'XXX.ITSO99.D1193.T0020546.S00022' CLUSTER
000023 DELETE 'XXX.ITSO99.D1193.T0020546.S00023' CLUSTER
000024 DELETE 'XXX.ITSO99.D1193.T0020546.S00024' CLUSTER
:continued see also next Figure
```

Figure 10-35  Clean up with data set IDCAMS delete statements for the work data sets
Figure 10-36 shows the results after you start the clean-up procedure.

Delete extend map data sets only if you are finished with all sessions. If other sessions are still active and you want to start cleaning the work data sets, remove the part of the jobs that deletes the Extend maps.

Do not forget to export the user catalog for the work data sets and delete all aliases for the work data sets.
Using TDMF TCP/IP for z/OS

This chapter describes the TDMF TCP/IP for z/OS product. It explains the additional steps in relation to a normal TDMF migration required to use the product in a typical remote migration scenario. The chapter also includes a general guidelines section.

This chapter contains the following sections:

- TDMF TCP/IP for z/OS overview
- TDMF general guidelines
- TDMF TCP/IP example replication
11.1 TDMF TCP/IP for z/OS overview

Transparent Data Migration Facility (TDMF) TCP/IP for z/OS is a host-based software solution for data replication. It allows migration in a z/OS environment to a new location over a long distance. TDMF TCP/IP for z/OS is user-initiated and controlled, and guarantees full access to the local data at any point during a replication operation.

11.1.1 Long-distance data migration software solution

TDMF TCP/IP for z/OS has the following advantages:

- When the Source or Target Storage System is not IBM storage, or uses different hardware. Metro Mirror, Global Copy, and other vendor copy services cannot be used or are not possible.
- If XRC / GM for z/OS cannot be installed, no HW resources for XRC feature (processor, cache, HBA, and so on) are available.
- Easy to manage: No additional automation or management is needed.
- Easy to install: No need to install additional HW
- Support any to any storage vendor, hardware independent

11.1.2 Long-distance data migration characteristics

TDMF TCP/IP for z/OS has the following characteristics:

- It allows long-distance data migration. This type of migration is the copying of data from one device (the source) to another device (the target) over an Ethernet connection using TCP/IP.
- A replication is the logical relationship between a source and target device.
- The user initiates and controls all migrations. The user identifies the “from” (source) volumes and the “to” (target) volumes.
- Multiple volume replication can be established during any single session.
- Multiple sessions run in parallel, and start sequentially.
- The replication tool is dynamically activated and manageable.
- Applications are unaware that replication is underway. The local data is continuously and fully accessible for read and write activity.
- For synchronization, all applications must be stopped at a certain point to get consistent data at the remote site.
- The tool supports a multiple-system shared data environment.
- The tool guarantees complete physical data integrity.
- The use of the tool is not restricted to any control unit model type or device type. Except as noted, all devices in the data center can participate in replication sessions as required.
- All volumes (source and target) of a replication must be online.
- No user can allocate data to a target volume during migration.

Restriction: TDMF for z/OS does not allow access to the target volume during the migration process.
A source volume cannot contain an active local page data set.
The source and target volumes must be of the same track geometry.

The proceeding characteristics represent the ideal of a non-disruptive data migration.

11.1.3 Terminology

These terms are used with the TDMF TCP/IP installation and replication process:

Master system (local) The local TDMF system. This system runs as an MVS batch job that is responsible for the data copy function. Unlike a local TDMF migration, there are two Master systems in one TDMF session: Local and remote.

Agent system (local) An associated TDMF MVS image running in a shared local storage environment with the Master. To ensure data integrity, any MVS LPAR that has access to the source volumes must be run using either the Master or an Agent system. The Master and associated Agent systems communicate through a shared system communications data set (COMMDS).

Master system (remote) The remote TDMF system. This system runs as an MVS batch job that is responsible for receiving and storing the data at the remote site.

Agent systems (remote) Agent systems are allowed at remote site. However, avoid sharing the remote volumes with other LPARs. TDMF prevents access by other users and systems to these volumes during the replication. If the target volumes are shared at remote site, Agents must be active at these systems.

Source volume DASD volumes containing the data to be replicated.
Target volume DASD volumes receiving the replicated data.
Migration pair The relationship between one source and one target volume for a single replication.
Migration group A group of volume pairs that use the same group name.
Migration session One Master (local) plus any Agents (local), one Master (remote), and pairs or groups to be processed in a single TDMF execution.
Data consistency The procedures to ensure that data remains intact and correct through the migration. All applications at the local site must be stopped before TDMF TCP/IP can go to synchronization to ensure consistent data.
Synchronization The time when I/O is inhibited to the source devices (quiesce phase), all applications stopped. During this time, the last of the updates are collected and reflected to the target devices (Synchronization phase). The Volid of the target volume is NOT changed during this process. It must be changed separately. When this process is complete, the TDMF session ends and the target device is marked offline (depends on replication parameter).
11.1.4 TDMF z/OS components architecture

Figure 11-1 depicts the TDMF z/OS architecture and components. In this example, the volumes are being moved from a local to a remote site. All storage vendors are supported (any to any).

The following components are depicted:

▶ TDMF TSO monitor: The TDMF TSO Monitor is used to manage the active replication jobs and view active or past replications. It is installed at the same time as the TDMF tool. The TDMF TSO Monitor Feature consists of REXX execs, which require ISPF version 3.6 and TSO/E Version 2.4 at minimum.

▶ Communications data set (local / remote): The local communication data set (COMMDS) is used to pass information between local systems participating in a TDMF TCP/IP session. This data set contains the status and messages related to a specific session. The COMMDS also serves as the input file to the TDMF TSO monitor. A COMMDS is also needed at the remote site, and need to have the same size (number of cylinders).

▶ Master and Agents: Each LPAR that has access to the storage subsystem and shares the source volumes must run an Agent. Each Master/Agent session must have access to its own communication data set. Multiple Agent systems can be involved in a session locally. The remote site normally has only a single Master and no Agents.

**Important:** To avoid a possible data integrity exposure, all systems accessing replicated volumes must be identified to the Master system. TDMF includes various controls and checks that ensure that the user does not make the following errors:

▶ Assign or direct conflicting migrations to the same devices
▶ Attempt migrations to nonexistent devices
▶ Use the same COMMDS for two simultaneous or overlapping migration sessions
Master system responsibilities
The TDMF Master system is responsible for the following tasks:

- Initialize the TDMF Master system environment and the COMMDS.
- Start and control each session for all participating systems.
- Check the communication with the remote site (remote master).
- Monitor source volume user I/O activity to detect updates.
- Copy data from the source volume to the target volume through TCP/IP.
- Process detected source volume updates from all systems.
- Perform refresh operations to the target site to reflect the update activity on the source volumes.
- Check the internal health of the Master environment (both sites) and the health of all Agent systems.
- The remote Master system receives data from local site and writes that data to the target volumes. It also monitors target volume system I/O activity to prevent updates from others to the target volumes.

Agent system responsibilities
The TDMF Agent system is responsible for the following items:

- Initialize the TDMF Agent environment and establish communications to the Master system through the COMMDS
- Acknowledge and process replication requests (volumes to be copied) from the Master system
- Monitor source volume user I/O activity and detect updates
- Notify the Master system of source volume update activity through the COMMDS
- Check the internal health of the Agent environments and the health of the Master system

11.1.5 TDMF z/OS hardware compatibility

TDMF is designed to support the count key data/extended (CKD/E)-capable control units of any vendors as a source or a target storage subsystem. This functionality includes Hyper-Volume, Flexible-Volumes, and all user-defined volume sizes. Extended address volumes (EAVs) might be supported depending on the z/OS version. For more information, see the TDMF Installation and reference manual.

Because TDMF is a host software migration tool, there are no particular hardware prerequisites.

11.1.6 Install TDMF TCP/IP z/OS

For the installation procedure, see Chapter 9, “Using TDMF for z/OS” on page 395.

You need to install TDMF at the remote site, but you do not need any license key to run TDMF TCP/IP.
11.1.7 TDMF z/OS customer requirements

TDMF TCP/IP for z supports all MVS-based operating systems currently supported by IBM. It is not suitable for use with native z/VM and VM/ESA, native VM/XA, native VSE/ESA, native VSE/XA, or DOS. The Operating Systems Support Matrix is at:


Tip: Perform periodic checks of the Required IBM Maintenance and Technical Information Bulletins (TIBs). These requirements must be implemented to ensure successful TDMF operation.

Required IBM Maintenance:

IBM Technical Support:

Technical Information Bulletins:

11.2 TDMF general guidelines

This section offers general guidelines when using TDMF TCP/IP z/OS.

11.2.1 Keeping current

It is important to make sure before any migration that your TDMF TCP/IP z/OS code is current and up-to-date.

- Bookmark the TDMF Technical Support website. It contains information to aid in the successful execution of TDMF TCP/IP.
- Register for automatic email notification. Registration ensures that you will be notified whenever the Technical Support website is updated. You will receive the actual link for registration with your license key (by mail).
- Get a Group ID (Lotus Notes). Click Notification Registry on the Technical Support website.
- Download each PTF as it becomes available. When a new PTF is released, the download links to previous PTFs are removed. The list fixes, however, continue to be available. PTFs can be installed at a later time.
- For a non-SMP/E installation, TDMF PTFs are not available and must be installed. The license key must be saved before installing.
- Manuals are periodically updated. To ensure that you have the most current information, check for new Technical Information Bulletins (TIBs). TIBs are typically issued between manual releases.
- Make sure that your TSO user ID is allowed to start TDMF TCP/IP and ICKDSF.
- Create a volume list. The target volumes must have the same number of cylinders as the source volume or more. Make sure that you do not have duplicates (VOLID, source, or target device) in your list. Mark your system-related volumes in this list (IPL, IODF, XCF DS, Local pages, PLPA & COMMON DS, and so on) so they are easy to find at the remote site.
Group volumes in relation to application data (storage groups) into the same session. You might need to group volumes by control units because of performance issues. Performance problems might be caused by the ESCON connection or HW with less internal bandwidth. Make sure that same volume is not defined in different groups.

The TCP/IP link speed between the two sites depends on the following factors:
- Number of TB to be replicated
- The time available for production move
- The need to shut down applications at the local site
- IPL at the remote site.

For more information, see 11.3.4, “TDMF TCP/IP performance example” on page 512.

Consult with staff in charge of the client network people to establish a range of port numbers that can be used for replication. The number of ports needed depends on total number of sessions. However, remember that one TDMF session needs two ports in sequence. Make sure that ports are not blocked by firewalls between the two sites.

Reserve a number of volumes for local communication data sets (COMMDS). Allocate no more than 3-4 COMMDS for each volumes to prevent performance issues when all sessions are running. The total number of COMMDS must be the same as number of sessions. Allocate the same number of COMMDS at the remote site.

Define available and active initiators (one initiator for each session) for both sites. The service ratio must be SYSSTC or similar for all TDMF jobs

Create local temporary Local Page data sets and do a page delete to the original local page data sets. TDMF TCP/IP does not copy active local page data sets. The temporary local page data sets are not copied to the remote site.

Establish naming conventions for the job names because all jobs must run in parallel at all LPARs. Naming conventions can help avoid confusion. For more information, see 11.3.2, “Naming conventions” on page 497.

Schedule the migration for a time when heavy system I/O load such as database reorganization and BATCH work can be stopped.

Normally replication of storage volumes is combined with a tape/library system replication. It is also sometimes done in combination with open storage systems and servers. Therefore, it is important to talk to the whole team responsible for the replication. This publication addresses only the disk part.

11.2.2 Creating a consistent data structure

To move an entire data center, you must have a consistent data structure. This section highlights the components of a consistent data structure. At the local site, you have one or more processors, and one or more storage systems. At the remote site, you have a minimum of one z/OS system to receive the data.

The data structure has the following components. Figure 11-2 on page 490 shows how these components relate to each other.

- Local system: The local system consists of all volumes in relation to the system. Examples include IPL, IODF, local page data sets, and system-related volumes.
- TDMF volumes: These volumes are TDMF libraries and COMMDS volumes. These volumes are not moved.
- Temporary local page data sets: These data sets are on volumes that are not replicated. TDMF cannot copy active local page data sets, so the original data sets are removed from the page pool by a page delete command. They can then be replicated.

- Data volumes: This term includes all volumes, including storage management system (SMS) and non-SMS managed volumes with applications, application data, and temporary work data sets.

- Remote system: This system includes all volumes at the remote site in relation to the remote master system. This system receives the data from local master over IP and writes the data to the target volumes in relation to the defined volume pairs.

- TDMF Monitor: This TDMF REXX ISPF application runs at a TSO user ID. This user must have access to all LPARs running TDMF Master/Agents sessions.

- TCP/IP Link: This link is the physical connection from local to remote site using TCP/IP as the protocol. The normal bandwidth for a service engagement (2-12 TB) is a 1 GBit link speed. With this connection, you can expect up to 120 MBps at the link.

### Figure 11-2  TDMF TCP/IP overview for a data center move to a new location

#### 11.2.3 Summary of replication tasks and steps

Before you begin the replication, perform the following steps:

- Do system assurance for both sites, including the TCP/IP connection. For more information, see the TDMF Installation and Reference Manual.

- Check that all volumes are available and online on the target system (new storage).

- Install TDMF TCP/IP on both sites with the systems programmer.

- Define the volume pairs (source / targets) with the storage administrator.
Create TDMF JOBs for both sites.
Create ICKDSF INIT / Rename Jobs for the remote site.
Prepare temporary local page data sets on the volumes that are not moved. These data sets are necessary because active local pages cannot be handled by TDMF.
Do a first network test to check the link, connection, and bandwidth.

The replication is normally done in three stages. Along with the storage administrator, update the volume list (relation source and target volumes) is updated to the actual status before each stage.

1. Perform a test without synchronization data as shown in Figure 11-3. This test is normally done with system/network volumes only, and is also a network test. Application data are copied to test the bandwidth.

![TDMF TCPIP – Sequence of Replication](image-url)

**Figure 11-3  Test without sync data**
2. Run a test with the complete synchronization data to test the system, network, and application data at remote site (Figure 11-4).

**Figure 11-4**  Test with synchronization data, Production Test Environment (PTE)
3. After completing the tests, run the actual production move (Figure 11-5). All data must be copied again because no updates were performed during the tests. In addition, there was no relation between the local and remote sites.

Figure 11-5  Production move, sequence of replication
### 11.2.4 Link performance information

Figure 11-6 shows the performance found in test environments for various connection types, and the expected throughput for TDMF TCP/IP.

<table>
<thead>
<tr>
<th>Connection Type</th>
<th>Bytes Per Minute</th>
<th>MB Per Minute</th>
<th>MB per Hour</th>
<th>GB per Hour</th>
<th>TDMF Speed GB per Hour</th>
</tr>
</thead>
<tbody>
<tr>
<td>384K DSL</td>
<td>2,949,120</td>
<td>2.8125</td>
<td>168.75</td>
<td>.1647</td>
<td>.1318</td>
</tr>
<tr>
<td>T1</td>
<td>11,520,000</td>
<td>10.98</td>
<td>658.18</td>
<td>.6437</td>
<td>.5149</td>
</tr>
<tr>
<td>10 base T Ethernet</td>
<td>75,000,000</td>
<td>71.52</td>
<td>4,291.53</td>
<td>4.19</td>
<td>3.35</td>
</tr>
<tr>
<td>T3</td>
<td>345,600,000</td>
<td>329.589</td>
<td>19,775</td>
<td>19.31</td>
<td>15.45</td>
</tr>
<tr>
<td>100 Base T Ethernet</td>
<td>750,000,000</td>
<td>715.255</td>
<td>42,915</td>
<td>41.90</td>
<td>33.52</td>
</tr>
<tr>
<td>OC3</td>
<td>1,162,500,000</td>
<td>1,108.64</td>
<td>66,518</td>
<td>65.95</td>
<td>51.96</td>
</tr>
<tr>
<td>OC12</td>
<td>4,500,000,000</td>
<td>4,291.53</td>
<td>257,492</td>
<td>251.45</td>
<td>201.16</td>
</tr>
<tr>
<td>Gigabit Ethernet</td>
<td>7,500,000,000</td>
<td>7,152.55</td>
<td>429,153</td>
<td>419.09</td>
<td>335.27</td>
</tr>
<tr>
<td>OC48</td>
<td>18,000,000,000</td>
<td>17,166.13</td>
<td>1,029,968</td>
<td>1,005.82</td>
<td>804.66</td>
</tr>
<tr>
<td>OC192</td>
<td>72,000,000,000</td>
<td>68,664.52</td>
<td>4,119,872</td>
<td>4,023.28</td>
<td>3,218.64</td>
</tr>
</tbody>
</table>

*Figure 11-6 Link speed examples*
Figure 11-7 shows performance information for an example involving moving 7 TB (1424 volumes) over 400 km with a dedicated 1 GBit link. The information was collected every 15 minutes from the IBM RMF™ monitor at the local and remote sites. The sessions were not started in sequence. Information from the test move showed that the best results came from putting the volumes/sessions with high update rates at the end of the migration windows.

Figure 11-7  List with RMFMON performance information
Figure 11-8 shows the impact on sessions running with software compression ON or OFF for the same example. With software compression ON, the link operated at not more than 120 MBps. However, the FICON channels at the remote site can achieve up to 160 MBps. In the example, software compression cannot be used during online business because of the high processor load. This setting cannot be changed dynamically.

**Figure 11-8  Example: Performance information client production move (TDMF TCP/IP 5.2)**

## 11.3 TDMF TCP/IP example replication

This example shows how to create TDMF, ICKDSF Rename, and ICKDSF INIT jobs. It includes an example of naming conventions. The network test is also included with screen captures of the JCL, monitor, and some performance information.

The distance in this example was around 100 km between local and remote site. The client wanted to test the connectivity and bandwidth for the two new defined links of 1 GBit each. To retain the performance information, the test was run twice and the resulting information was in different COMMDS. Sessions 01 - 03 were in the first test, and sessions 04 - 06 were in the second.

Both tests used the same 30 volumes (10 out of each DISK Storage unit). Therefore, after the first test the RENAME and INIT jobs were run, and all 30 volumes were set ONLINE again. All devices with 2xxx and 3xxx addresses were attached with ESCON Channel paths. The rest were FICON attached. Example 11-1 shows the list of volumes there were used.

**Example 11-1  Volume list for test**

<table>
<thead>
<tr>
<th>Source volumes at source site</th>
<th>Target volumes at remote site</th>
</tr>
</thead>
<tbody>
<tr>
<td>UC2010 - UC2019 (ESCON connection)</td>
<td>UC8400 - UC8409 (FICON connection)</td>
</tr>
<tr>
<td>UC3010 - UC3019 (ESCON connection)</td>
<td></td>
</tr>
<tr>
<td>UC4010 - UC4019 (FICON connection)</td>
<td></td>
</tr>
</tbody>
</table>
11.3.1 Overview of replication

Figure 11-9 shows a simplified view of the physical setup of the example. The Channel Path IDs (CHP) are listed to check the performance using the performance monitor (RMFMON) of z/OS.

11.3.2 Naming conventions

Before creating the JCLs, create naming conventions to avoid confusion during the migration process. The following is an example from a service project.

These sessions require one master and six agents for each session. Therefore, there are seven LPARs. Two tests are run with three sessions each to test different links. All of the information is stored in the COMMDS.

In total, 42 jobs (3 (sessions) * 7 (# of LPARs) * 2 (tests)) need to be created.
The naming conventions used are shown in Example 11-2.

Example 11-2  Job naming convention (example)

//TDMLCPO1 JOB ..... 
!!! !
!!! +--> session number (2 digits)
!!+----> 2 char out of the LPAR name (first or last 2 char)
!! to identify AGENTs at different LPARs, make sure not have
!! duplicate jobnames, only the first one will start.
!+-----> L=Local site, R=Remote site
+------- M=Master, A=AGENT
JCL examples at remote site

The job names generated at the local site are shown in Figure 11-10 on page 499.

<table>
<thead>
<tr>
<th>Name</th>
<th>Prompt</th>
<th>Size</th>
<th>Created</th>
<th>Changed</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>$OVR00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>--&gt; Overview</td>
</tr>
<tr>
<td>TDALCD01</td>
<td>Agents for SYCD</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDALCD02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDALCD03</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDALCD04</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDALCD05</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDALCD06</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDALCR01</td>
<td>Agents for SYCR</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDALCR02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDALCR03</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDALCR04</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDALCR05</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDALCR06</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDALFD01</td>
<td>Agents for SYFD</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDALFD02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDALFD03</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDALFD04</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDALFD05</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDALFD06</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDALFP01</td>
<td>Agents for SYFD</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDALFP02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDALFP03</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDALFP04</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDALFP05</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDALFP06</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDAL0601</td>
<td>Agents for SY06</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDAL0602</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDAL0603</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDAL0604</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDAL0605</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDAL0606</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDAL0701</td>
<td>Agents for SY07</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDAL0702</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDAL0703</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDAL0704</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDAL0705</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDAL0706</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDL$G00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>&lt;-- create COMMDS</td>
</tr>
<tr>
<td>TDMTCP01</td>
<td>Master for SYCP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDMTCP02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDMTCP03</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDMTCP04</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDMTCP05</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDMTCP06</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>End</strong></td>
</tr>
</tbody>
</table>

Figure 11-10  Job names in our test replication (local site)
In a large environment, it can be hard to find the jobs that belong together. However, since z/OS 1.8 there is a new function, build in ISPF, to set a filter for a partitioned data set extended (PDSE). This filter function is shown in Figure 11-11.

![Figure 11-11 Filter function for PDSE members](image)

The result of this filter command is shown in Figure 11-12. All members for session 01 are displayed, making it easy to submit the right jobs.

![Figure 11-12 Display of PDSE members fit to the filter criteria (*01) only](image)
For long-distance data migration, you must specify a new parameter for replication to address
the remote master. Specify an IP address and a port number as shown on line 22 in
Figure 11-13.

```
EDIT       MBSM.TDMF.NWTS1.TDMF.CNTL(TDMLCP01) - 01.00  Columns 00001 00072
Command ===>                                                  Scroll ===> CSR
****** **************************************** Top of Data ****************************
000001 //TDMLCP01 JOB (,DASD), 'IBM DATA MIGRATION', CLASS=T,
000002 // REGION=OM, MSGCLASS=X, NOTIFY=&SYSUID
000003 /* TDMF DATA MIGRATION - DO NOT CANCEL THIS JOB !!!
000004 /* telefon number IBM service person 1
000005 /* telefon number IBM service person 2
000006 /*XEQ TXRPRD
000007 /*JOBPARM SYSAFF=SYCP
000008 /******************************************************************************
000009 /// Master JOB TDMF running: SYCP local site
000010 ///*****************************************************************************
000011 //LOCAL01 EXEC PGM=GTDMAIN, PARM=MASTER, REGION=0M
000012 //STEPLIB DD DISP=SHR, DSN=SYS1.GTD530.GTDLLIB
000013 //GTDKEY DD DISP=SHR, DSN=SYS1.GTD530.GTDLLIB
000014 //SYSCOM DD DISP=SHR, DSN=SYS1.TDMF.SDNT.LOCAL.SYSCOM01
000015 //SYSPRINT DD SYSOUT=* 000016 //SYSSNAP DD SYSOUT=* 000017 //SYSSIN DD *
000018 SESSION M(DG72)
000019 AGENTS( DH72 DG72 DI72 DK72 GA98 GB98)
000020 OPT(FAST UNIDENT(N) PROMPT CONC(4 ACTIVE)
000021 NOCOMPRESS PAC CHECKT OFFLINET SINGLE)
000022 REMOTE DX72 ADDR(192.168.2.12) PORT(8200)
000023 REPlicate UC2010 UC8400 OPT(MAXTR(1))
000024 REPlicate UC2011 UC8401 OPT(MAXTR(1))
000025 REPlicate UC2012 UC8402 OPT(MAXTR(1))
000026 REPlicate UC2013 UC8403 OPT(MAXTR(1))
000027 REPlicate UC2014 UC8404 OPT(MAXTR(1))
000028 REPlicate UC2015 UC8405 OPT(MAXTR(1))
000029 REPlicate UC2016 UC8406 OPT(MAXTR(1))
000030 REPlicate UC2017 UC8407 OPT(MAXTR(1))
000031 REPlicate UC2018 UC8408 OPT(MAXTR(1))
000032 REPlicate UC2019 UC8409 OPT(MAXTR(1))
****** **************************************** Bottom of Data ****************************
```

Figure 11-13  TDMF master JCL local site (TDMLCP01)
In Figure 11-14 shows an example for a local Agent job.

```
EDIT               MBSM.TDMF.NWTS1.TDMF.CNTL(TDALCD01) - 01.00  Columns 00001 00072
Command ====>     Scroll ====> CSR

000001  //TDALCD01 JOB (,DASD),'IBM DATA MIGRATION',CLASS=T,
000002  //      REGION=0M,MSGCLASS=X,NOTIFY=&SYSUID
000003  // TDMF DATA MIGRATION - DO NOT CANCEL THIS JOB !!!
000004  //* telephone number IBM service person 1
000005  //* telephone number IBM service person 2
000006  /*XEQ TORCSD
000007  /*JOBPARM SYSAFF=SYCD
000008  /********************************************************************************
000009  //* AGENT JOB TDMF running: SYCD
000010  /********************************************************************************
000011  //STEP01 EXEC PGM=GTDMAIN,PARM=AGENT,REGION=OM
000012  //STEPLIB DD DISP=SHR,DSN=SYS1.GTD530.GTDLLIB
000013  //GTDKEY   DD DISP=SHR,DSN=SYS1.GTD530.GTDLLIB
000014  //SYSCOM   DD DISP=SHR,DSN=SYS1.TDMF.SDNT.LOCAL.SYSCOM01
000015  //SYSPRINT DD SYSOUT=*  
000016  //SYSSDUMP DD SYSOUT=*  
000017  //SYSSNAP  DD SYSOUT=*  

Figure 11-14  TDMF Agent JCL local site (TDALCD01)
```

The following are other parameters used in this example:

- **FAST**
  Copy allocated space only.

- **UNIDENT(W)**
  Check for other systems without an agent, and send a warning if one is found. For a test or production move, use UNIDENT(T) so if an unknown system is available, the session terminates.

- **PROMPT**
  Stop at REFRESH status and wait for confirmation before going to sync.

- **CONC(4 ACTIVE)**
  Run four volumes concurrently. This value can be changed dynamically later using the TDMF monitor.

- **NOCOMPRESS**
  Do not use software compression.

- **PAC**
  Use pacing to automatically reduce the number of tracks per I/O during a performance problem.

- **CHECKT**
  Check the target volume for data, and stop the migration if data are allocated there.

- **OFFLINET**
  After session ends, set the target volumes offline.

- **SINGLE**
  Combines all volumes into a single session that you can end (go to sync) using a manual command.

- **MAXTRACKS(15)**
  This parameter adjusts the number of tracks read per I/O and the memory usage in the common area. If you run out of memory, use MAXTR(1) to reduce the performance impact. This parameter is also useful for old devices attached using ESCON.

For more information about the parameters, see the TDMF Installation and Reference Manual.
JCL examples at remote site

The remote site requires an additional JCL. The migration then proceeds with the following steps:

1. The TDMF master jobs receive the data and write it to the target volumes. The list used in the example is shown in Figure 11-15.

```
<table>
<thead>
<tr>
<th>Command - Enter &quot;/&quot; to select action</th>
<th>Message</th>
<th>Volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>MBSM.TDMF.NWTS1.INIT.CNTL</td>
<td>TDM001</td>
<td></td>
</tr>
<tr>
<td>MBSM.TDMF.NWTS1.RENAME.CNTL</td>
<td>TDM001</td>
<td></td>
</tr>
<tr>
<td>MBSM.TDMF.NWTS1.TDMF.CNTL</td>
<td>TDM001</td>
<td></td>
</tr>
<tr>
<td>************************************</td>
<td>End of Data Set list ************************************</td>
<td></td>
</tr>
</tbody>
</table>
```

*Figure 11-15  TDMF and ICKDSF PDSE list used for Network test*

The remote site is shown in Figure 11-16.

```
<table>
<thead>
<tr>
<th>Name</th>
<th>Prompt</th>
<th>Size</th>
<th>Created</th>
<th>Changed</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>$OVR00</td>
<td>TDCR$G00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDMRX101</td>
<td>Master for SYX1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDMRX102</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDMRX103</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDMRX104</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDMRX105</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDMRX106</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VDASIN01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
*Figure 11-16  TDMF Master jobs remote site*

Tip: Do not define the source or target volumes at the remote site. The port number in relation to the local session port number is important. The remote master listens to the TCP/IP port defined in the JCL. As soon the TCP/IP session is established, this port is used to exchange the volume information. The following port number (port 8201) is used at the remote site to establish for each source/target volume pair an own TCP/IP session. See Figure 11-34 on page 513.
The Master JCL remote site is shown in Figure 11-17.

```
EDIT       MBSM.TDMF.NWTS1.TDMF.CNTL(TDMR0601) - 01.00  Columns 00001 00072
Command ====>  Scroll ====> CSR
****** *************** Top of Data ***********************
000001 //TDMRX101 JOB (,DASD),'IBM DATA MIGRATION',CLASS=T,
000002  //   REGION=OM,MSGCLASS=X,NOTIFY=&SYSUID
000003 //  TDMF DATA MIGRATION - DO NOT CANCEL THIS JOB !!!
000004 //  telefon number   IBM service person 1
000005 //  telefon number   IBM service person 2
000006  //  *XEQ SYX1
000007 /*JOPARM SYSAFF=SYX1
000008  *****************************************%
000009 //  ** MASTER JOB TDMF SYX1 remote site
000010  *******************************************%
000011 //STEP01 EXEC PGM=GTDMAIN,PARM='MASTER,PORT=8200'
000012 //STEPLIB DD DISP=SHR,DSN=SYS1.GTD530.GTDLLIB
000013 //GTDKEY   DD DISP=SHR,DSN=SYS1.GTD530.GTDLLIB
000014 //SYSCOM   DD DISP=SHR,DSN=SYS1.TDMF.SDNT.REMOTE.SYSCOM01
000015 //SYSPRINT DD SYSOUT=*  
000016 //SYSDUMP DD SYSOUT=*  
000017 //SYSSNAP DD SYSOUT=*  
000018 //SYSSNAP DD SYSOUT=*  
000019 SESSION M(DX72)
000020  OPT(FAST UNIDENT(T))
****** **************** Bottom of Data ********************
```

*Figure 11-17  TDMF Master JCL remote site*

2. After replication, rename the target volumes to their original VOLID. Otherwise, the client cannot work with these volumes for test. To rename the volumes, use ICKDSF REFORMAT, which is a z/OS system tool (Figure 11-18).

```
EDIT              MBSM.TDMF.NWTS1.RENAME.CNTL  Row 00001 of 00006
Command ==>  Scroll ==> CSR
Name  Prompt  Size  Created  Changed  ID
      ____________  _______  _______  _______  _______
      TDRR$G01
      TDRR$G02
      TDRR$G03
      TDRR$G04
      TDRR$G05
      TDRR$G06
      **End**
```

*Figure 11-18  ICKDSF RENAME jobs*
The results of the rename job on session 01 are shown in Figure 11-19.

```
EDIT       MBSM.TDMF.NWTS1.RENAME.CNTL(TDRR$G01) - 01.0 Columns 00001 00072
Command ===>                                                  Scroll ===> CSR
****** ************************************************************** Top of Data **************************************************************
000001 //TDRR$G01 JOB (,DASD),'IBM DATA MIGRATION',CLASS=T,
000002 //      REGION=0M,MSGCLASS=X,NOTIFY=&SYSUID
000003 //** TDMF DATA MIGRATION - DO NOT CANCEL THIS JOB !!!
000004 //** telefon number IBM service person 1
000005 //** telefon number IBM service person 2
000006 //STEP0001 EXEC PGM=ICKDSF,REGION=3000K,PARM='NOREPLYU'
000007 //SYSPRINT DD SYSOUT=(*,,1TAB)
000008 //SYSIN DD *
000009 REFORMAT UNIT(8400) VFY(UC8400) VOLID(UC2010)
000010 REFORMAT UNIT(8401) VFY(UC8401) VOLID(UC2011)
000011 REFORMAT UNIT(8402) VFY(UC8402) VOLID(UC2012)
000012 REFORMAT UNIT(8403) VFY(UC8403) VOLID(UC2013)
000013 REFORMAT UNIT(8404) VFY(UC8404) VOLID(UC2014)
000014 REFORMAT UNIT(8405) VFY(UC8405) VOLID(UC2015)
000015 REFORMAT UNIT(8406) VFY(UC8406) VOLID(UC2016)
000016 REFORMAT UNIT(8407) VFY(UC8407) VOLID(UC2017)
000017 REFORMAT UNIT(8408) VFY(UC8408) VOLID(UC2018)
000018 REFORMAT UNIT(8409) VFY(UC8409) VOLID(UC2019)
****** ************************************************************** Bottom of Data **************************************************************
```

Figure 11-19  ICKDSF RENAME job example for target volumes in session "01"

3. The ICKDSF init jobs initialize the target volumes back to their old VOLID. Therefore, you do not have to change the TDMF JCL again (Figure 11-20).

```
EDIT       MBSM.TDMF.NWTS1.INIT.CNTL                              Row 00001 of 00006
Command ===>                                                  Scroll ===> CSR
Name        Prompt    Size   Created          Changed     ID
_________    _________   ____   ----------          --------    
        TDMI$G01
        TDMI$G02
        TDMI$G03
        TDMI$G04
        TDMI$G05
        TDMI$G06
**End**
```

Figure 11-20  ICKDSF INIT jobs
The results of the init jobs on the target volumes is session 01 are shown in Figure 11-21.

```
EDIT MBSM.TDMF.NWTS1.INIT.CNTL(TDMI$G01) - 01.00 Columns 00001 00072
Command ==>  Scroll ==> CSR
****** *************************************************** Top of Data *********************************************
000001 //TDMI$G01 JOB (.DASD),'IBM DATA MIGRATION',CLASS=T,
000002 // REGION=OM,MSGCLASS=X,NOTIFY=&SYSUID
000003 /* TDMF DATA MIGRATION - DO NOT CANCEL THIS JOB !!!
000004 /* telefon number IBM service person 1
000005 /* telefon number IBM service person 2
000006 //STEP0001 EXEC PGM=ICKDSF,REGION=3000K,PARM='NOREPLYU'
000007 //SYSPRINT DD SYSOUT=(*,,1TAB)
000008 //SYSPRINT DD * 
000009 INIT UNIT(8400) VFY(UC2010) VOLID(UC8400)
000010 INIT UNIT(8401) VFY(UC2011) VOLID(UC8401)
000011 INIT UNIT(8402) VFY(UC2012) VOLID(UC8402)
000012 INIT UNIT(8403) VFY(UC2013) VOLID(UC8403)
000013 INIT UNIT(8404) VFY(UC2014) VOLID(UC8404)
000014 INIT UNIT(8405) VFY(UC2015) VOLID(UC8405)
000015 INIT UNIT(8406) VFY(UC2016) VOLID(UC8406)
000016 INIT UNIT(8407) VFY(UC2017) VOLID(UC8407)
000017 INIT UNIT(8408) VFY(UC2018) VOLID(UC8408)
000018 INIT UNIT(8409) VFY(UC2019) VOLID(UC8409)
****** *************************************************** Bottom of Data *********************************************
```

Figure 11-21  ICKDSF INIT job example for target volumes in TDMF session "01"

Rename the volumes to their original VOLIDs after replication so you can start tests. Before the next test or the production move, you must switch it back. Use the ICKDSF verify (VFY) parameter in combination with NOREPLY=Y, so you can run this job without a manual REPLY at the operator console.
11.3.3 TDMF TCP/IP examples for a network test

The following shows the first session and what is going on in the TDMF Monitor and the RMF performance monitor.

1. Start the local Master and local Agents as shown in Figure 11-22.

   TDMF is waiting now for the remote master. As soon the TDMF remote master is started, TDMF starts the copy process for the defined concurrent number of volumes.

   After a few seconds, you see the estimated copy phase end time behind each volume as shown in Figure 11-23.

   ![Figure 11-22 TDMF Monitor Item 1 (local session started)](image)

   ![Figure 11-23 TDMF Monitor, start copy phase](image)
2. If there is no impact to the production, increase the concurrent number of volumes by switching to TDMF Monitor Item 2. Dynamically change the number of volumes running concurrently for example to 8 as shown in Figure 11-24.

```
Figure 11-24   TDMF menu item 2, change number of concurrent volumes
```

3. Switch back to TDMF menu item 1 pressing F10. There are now four additional running volumes, for a total of eight (Figure 11-25).

```
Figure 11-25   TDMF menu item 1
```
After a few minutes, you see the window shown in Figure 11-26.

![Session Monitor](image1)

4. When the initial copy phase is done, you see the updates (here Refresh x) and TDMF refreshes the volumes asynchronously. The message **SYNC Volume Needed** means that this session is ready to go to SYNC (Figure 11-27).

![Load is running to the source volumes](image2)

**Figure 11-26  TDMF menu item 1**

**Figure 11-27  TDMF menu item 1, session in refresh phase**
In this example, the updates to the source volumes were stopped to emulate an application stop. Start the SYNC phase using the PG command in front of the first volume (Figure 11-28). You must confirm the command at the request window.

```
<table>
<thead>
<tr>
<th>Requested</th>
<th>Volume Device</th>
<th>Group</th>
<th>--- Migration ---</th>
<th>- Error Info -</th>
<th>Sync</th>
</tr>
</thead>
<tbody>
<tr>
<td>pg</td>
<td>*UC2010</td>
<td>2010</td>
<td>SINGLE</td>
<td>Waiting Sync</td>
<td>PIT</td>
</tr>
<tr>
<td></td>
<td>UC8400</td>
<td>192.168.002.012 Port=08200</td>
<td></td>
<td></td>
<td>005</td>
</tr>
<tr>
<td></td>
<td>__</td>
<td>UC2011</td>
<td>2011</td>
<td>SINGLE</td>
<td>Refresh 1</td>
</tr>
<tr>
<td></td>
<td>__</td>
<td>UC8401</td>
<td>192.168.002.012 Port=08200</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>__</td>
<td>UC2012</td>
<td>2012</td>
<td>SINGLE</td>
<td>Refresh 1</td>
</tr>
<tr>
<td></td>
<td>__</td>
<td>UC8402</td>
<td>192.168.002.012 Port=08200</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>__</td>
<td>UC2013</td>
<td>2013</td>
<td>SINGLE</td>
<td>Refresh 2</td>
</tr>
<tr>
<td></td>
<td>__</td>
<td>UC8403</td>
<td>192.168.002.012 Port=08200</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

*Figure 11-28  TDMF Monitor menu item 2, stop the session, go to SYNC, line command “PG”*
TDMF starts the SYNC phase as shown in Figure 11-29. The upper left area displays the total numbers of volumes in this session. Below this number, a counter displays the increasing number of completed volumes.

As soon all volumes within the session are completed, the session is no longer displayed in the TDMF monitor. The jobs belonging to this session end. Check all job log to see whether MAXCC is greater than zero (Figure 11-30).

For the local master job MAXCC was equal to four as shown in Figure 11-31. All warning messages result in a MAXCC=4. In this test, there was an additional LPAR with access to the replicated source volumes. No Agent was started at this LPAR because the source volumes were OFFLINE.

If MAXCC is greater than four, check the job logs for error messages. You will need to restart the session.
11.3.4 TDMF TCP/IP performance example

In this test, RMFMON was run and displayed the channel load for the Open System adapter (OSA) CHPID 28 (Figure 11-32).

There was not more than 64 MBps at the OSA connection. Because the source volumes were ESCON attached, there might not be more than 15 M/sec each channel path. There were four paths to the control unit plus TCP/IP usage.

<table>
<thead>
<tr>
<th>ID</th>
<th>NO</th>
<th>G</th>
<th>TYPE</th>
<th>S</th>
<th>PART</th>
<th>TOT</th>
<th>BUS</th>
<th>PART</th>
<th>TOT</th>
<th>PART</th>
<th>TOT</th>
<th>Rate</th>
<th>ACTV</th>
<th>Rate</th>
<th>ACTV</th>
</tr>
</thead>
<tbody>
<tr>
<td>28</td>
<td>OSD</td>
<td>Y</td>
<td>46.5</td>
<td>47.2</td>
<td>35.6</td>
<td>227K</td>
<td>229K</td>
<td>64M</td>
<td>64M</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>OSE</td>
<td>Y</td>
<td>0.0</td>
<td>0.3</td>
<td>7.3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(Figure 11-32) RMFMON Channel statistics

11.3.5 TCP/IP information

This section covers information about the TCP/IP connections

TCP/IP settings for TDMF replication

The following TCP/IP settings must be checked and updated for test/production move. The TCP/IP send and receive buffers should have these values listed in Figure 11-33, which are 256 KB each.

You can view this information using the TSO netstat config command.

(EZZ2350I) MVS TCP/IP NETSTAT CS V1R11 TCP/IP Name: TCPIPBC 04:52:02
EZZ2700I Home address list:
EZZ27011 Address Link Flg
EZZ27021 ------ ---- ---
EZZ27031 192.168.2.04 LOSAEG09 P
EZZ27031 127.0.0.1 LOOPBACK
EZZ2350I MVS TCP/IP NETSTAT CS V1R11 TCP/IP Name: TCPIPBC 04:52:02
EZZ2720I TCP Configuration Table:
EZZ2721I DefaultRcvBufSize: 00262144 DefaultSndBufSize: 00262144
EZZ2727I DefltMaxRcvBufSize: 00262144 SoMaxConn: 0000000010
EZZ2722I MaxReTransmitTime: 120.000 MinReTransmitTime: 0.500
EZZ2723I RoundTripGain: 0.125 VarianceGain: 0.250
EZZ2724I VarianceMultiplier: 2.000 MaxSegLifeTime: 30.000
EZZ2725I DefaultKeepAlive: 00000120 DelayAck: No
EZZ2726I RestrictLowPort: Yes SendGarbage: No
EZZ2728I TcpTimeStamp: Yes FinWait2Time: 300
EZZ2729I TTLs: No

Figure 11-33 Result of TCP/IP netstat config command

TCP/IP port mapping for a TDMF session

For each TDMF TCP/IP session, you need two ports at the remote site. The defined port in the TDMF Master JOBS (local and remote site) is used to establish access to the other site. It is also used for communication between the two master jobs.
In this example, the port numbers shown in Figure 11-34 are used. Each volume pair (source / target volume) has a TCP/IP session. At the local site, TCP/IP is using a free port mapped to the port 8201 for each volume pair. In the TDMF example, there are 10 volume pairs in the TDMF session. Therefore, you see 10 different ports in relation to port 8201 at the remote site transferring the data. There is also one port in relation to port 8200 at the remote site for the communication between the two master sessions.

To see these TCP/IP sessions, use the TSO `netstat conn` command.

```plaintext
Figure 11-34   Viewing the TCP/IP session port mapping for a TDMF session
```

<table>
<thead>
<tr>
<th>EZZ2350I</th>
<th>MVS TCP/IP NETSTAT CS VIR11</th>
<th>TCPIP Name: TCPIPCP</th>
<th>04:52:02</th>
</tr>
</thead>
<tbody>
<tr>
<td>EZZ2585I</td>
<td>User Id Conn Local Socket</td>
<td>Foreign Socket</td>
<td>State</td>
</tr>
<tr>
<td>EZZ2586I</td>
<td>----------------------------</td>
<td>---------------------</td>
<td>----------</td>
</tr>
<tr>
<td>EZZ2587I</td>
<td>BPXOINIT 00000013 0.0.0.0..10007</td>
<td>0.0.0.0..0</td>
<td>Listen</td>
</tr>
<tr>
<td>EZZ2587I</td>
<td>DB1ADIST 0000349D 0.0.0.0..5051</td>
<td>0.0.0.0..0</td>
<td>Listen</td>
</tr>
<tr>
<td>EZZ2587I</td>
<td>DB1ADIST 0000349E 0.0.0.0..5054</td>
<td>0.0.0.0..0</td>
<td>Listen</td>
</tr>
<tr>
<td>EZZ2587I</td>
<td>DB1ADIST 000034A2 0.0.0.0..5052</td>
<td>0.0.0.0..0</td>
<td>Listen</td>
</tr>
<tr>
<td>EZZ2587I</td>
<td>FTPDB1 00013FE7 0.0.0.0..21</td>
<td>0.0.0.0..0</td>
<td>Listen</td>
</tr>
<tr>
<td>EZZ2587I</td>
<td>TDMLBC01 000587A7 192.168.2.04..2230</td>
<td>192.168.2.12..8201</td>
<td>Establ</td>
</tr>
<tr>
<td>EZZ2587I</td>
<td>TDMLBC01 000587A8 192.168.2.04..2231</td>
<td>192.168.2.12..8201</td>
<td>Establ</td>
</tr>
<tr>
<td>EZZ2587I</td>
<td>TDMLBC01 000587AF 192.168.2.04..2232</td>
<td>192.168.2.12..8201</td>
<td>Establ</td>
</tr>
<tr>
<td>EZZ2587I</td>
<td>TDMLBC01 000587AE 192.168.2.04..2233</td>
<td>192.168.2.12..8201</td>
<td>Establ</td>
</tr>
<tr>
<td>EZZ2587I</td>
<td>TDMLBC01 000587A9 192.168.2.04..2234</td>
<td>192.168.2.12..8201</td>
<td>Establ</td>
</tr>
<tr>
<td>EZZ2587I</td>
<td>TDMLBC01 000587AC 192.168.2.04..2235</td>
<td>192.168.2.12..8201</td>
<td>Establ</td>
</tr>
<tr>
<td>EZZ2587I</td>
<td>TDMLBC01 000587AA 192.168.2.04..2236</td>
<td>192.168.2.12..8201</td>
<td>Establ</td>
</tr>
<tr>
<td>EZZ2587I</td>
<td>TDMLBC01 000587B7 192.168.2.04..2198</td>
<td>192.168.2.12..8200</td>
<td>Establ</td>
</tr>
<tr>
<td>EZZ2587I</td>
<td>TDMLBC01 000587AD 192.168.2.04..2237</td>
<td>192.168.2.12..8201</td>
<td>Establ</td>
</tr>
<tr>
<td>EZZ2587I</td>
<td>TDMLBC01 000587AB 192.168.2.04..2238</td>
<td>192.168.2.12..8201</td>
<td>Establ</td>
</tr>
<tr>
<td>EZZ2587I</td>
<td>TDMLBC01 000587A6 192.168.2.04..2229</td>
<td>192.168.2.12..8201</td>
<td>Establ</td>
</tr>
</tbody>
</table>

Figure 11-34   Viewing the TCP/IP session port mapping for a TDMF session
Network block devices for Linux

This appendix addresses techniques where block devices are presented to a host, but are physically attached to another host. This configuration allows you to migrate data in a host-based manner for Linux operating systems.

This appendix contains the following sections:

- Network block devices
- Summary
Network block devices

Network block devices are block devices where the real device is attached to a different Linux host. The device accepts the same block commands to write, read, and maintain data as a conventional disk. However, the block device forwards the I/Os through a network using TCP or UDP protocols to a server where the actual device is connected to.

This methodology allows you to migrate data per blocks to a different destination. Depending on the implementation, migrations can use low-level copy commands like `dd` or `cpio`. Some implementations allow an RAID-1 function where the data is mirrored automatically to two different network block devices.

The Linux Network Block Device `ndb` is a basic implementation of a generic network block device. It has been included in the kernel since version 2.1.101. It provides read-only and read/write access to a remote device. Many users mount the remote device as read only. In this configuration, only one host is allowed write access the remote device. However, there is no access control included in the code. This configuration carries the risk that data can be deleted if more than one client tries to write to this network device.

When the ndb server is started, it listens to a TPC or UDP port assigned by the administrator. The server is started with the command shown in Example A-1.

```
Example A-1 Starting the ndb-server

# ndb-server 5000 /dev/sda1
# _
```

In this example, the ndb device is represented by a file in the directory `/export`. When the server starts, it listens to port 5000 for incoming requests.

After the server is started, you can create a block device by referencing the IP address of the server and the port where the server listens (Example A-2).

```
Example A-2 Create a ndb device

# ndb-client 9.155.33.67 5000 /dev/ndb0
# _
```

The network block device can be read to create a file system, mount it, and allocate files and directories. For data migrations, however, you need to use other methods.

Suggested methods

Creating a file system in the network block device and copying the data file by file is a basic migration technique. A more efficient method is to copy the whole device in a single `dd` statement to the network device as shown in Example A-3.

```
Example A-3 Copying a raw device to the nbd

# dd if=/dev/sda2 of=/dev/nbd0 bs=1024
# _
```

Using method means that the applications using the data on the disk must be shut down. In addition, the file system must be unmounted to capture all recent I/Os from the cache. Depending on the used capacity of the disk and the bandwidth of the network, this process might take a while.
In the following sections, other processes that reduce the downtime because the synchronization can be done in the background are addressed. With these methods, the applications only need to be stopped for the cut over to the new storage.

Using software RAID1

The easiest way to replicate the data to the network device is to establish an RAID1 as a software RAID using the tool `mdadm`. The `mdadm` tool can provide different RAID levels using real block devices. Data migration using RAID1 involves the following steps:

1. Implement RAID1, which is mirroring between two volumes. After it is created, both volumes are synchronized in the background.
2. When the synchronization is completed, all updates by applications or users are replicated to both devices.
3. The migration is finalized when the file system is unmounted and you close RAID1 using the `mdadm` tool.

The following steps show a simple and straightforward example:

1. Close all applications and unmount the file system.
2. When the RAID1 device is established, a new device is allocated that writes the data to both devices. Mount the file system to this device as shown in Example A-4.

   **Example A-4   Create the RAID1 out of the local disk and the nbd0 device**

   ```
   # mdadm --create /dev/md0 --level=mirror --raid-devices=2 /dev/sda2 /dev/nbd0
   mdadm: /dev/sda2 appears to contain an ext2fs file system
   size=1048576K  mtime=Thu Jul  7 14:30:34 2011
   Continue creating array? y
   mdadm: array /dev/md0 started.
   #
   ```

3. The device `/dev/sda2` is the source which must be transmitted to the network block device `/dev/nbd0`. The result is a new device `/dev/md0`, which is now the RAID1 device.
4. The synchronization of both mirrors starts immediately. The progress of the synchronization can be monitored with the command shown in Example A-5.

   **Example A-5   Query the status of the synchronization**

   ```
   # cat /proc/mdstat
   Personalities : [raid1]
   md0 : active raid1 nbd0[1] dm-3[0]
       1048512 blocks [2/2] [UU]
           [==========>...............] resync = 31.2% (327680/1048512) finish=0.5min
       speed=23405K/sec
   unused devices: <none>
   #
   ```

   When the synchronization is complete, you see the status shown in Example A-6.

   **Example A-6   Status, when synchronization is done**

   ```
   # cat /proc/mdstat
   Personalities : [raid1]
   md0 : active raid1 nbd0[1] dm-3[0]
   ```
1048512 blocks [2/2] [UU]

unused devices: <none>

5. Mount the file system for the applications after RAID1 is established (Example A-7).

Example A-7  Mount the RAID1 device

```
# mount /dev/md0 /mnt
# df
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/sda1             14761044   8922208   5089004  64% /
none                   4071508       412   4071096   1% /dev
none                   4076096      1932   4074164   1% /dev/shm
none                   4076096     188   4075908   1% /var/run
none                   4076096         0   4076096   0% /var/lock
none                   4076096         0   4076096   0% /lib/init/rw
/dev/md0               1032088    103468    876192  11% /mnt   <===
```

6. Any updates from users or the applications are mirrored to both underlaying devices. When you are ready for the cut over, close the applications and unmount the file systems, and then close RAID1 as shown in Example A-8.

Example A-8  Cut over to the target device

```
# umount /mnt
#
# mdadm --stop /dev/md0
mdadm: stopped /dev/md0
#
# cat /proc/mdstat
Personalities : [raid1]
unused devices: <none>
```

7. When RAID1 is established, a md superblock is written to each device. To mount the remote device on the target system, remove this superblock as shown in Example A-9.

Example A-9  Clean up the superblock at the target system

```
# mdadm --zero-superblock /dev/nbd0
#
# mount /dev/sda1 /APPLICATION
```

8. The migration is now complete. Stop the nbd-server at the target system, stop the nbd-client at the source system, and clean up RAID1.

Summary

The Network Block Device nbd can be used for basic migration to new Linux systems and storage devices at different sites. This migration technique is effective for migrations involving only a few disks.
However, two major issues must be considered first:

- The communication between the server and the client is realized by specifying a TCP user port. Using a port might be a security issue in some environments.

- There is no control mechanism that prevents more than one client from writing data to the server. Having no control mechanism creates a high risk of corrupting the data on the server. You must ensure that no other client has access to the server device.
This chapter describes how to migrate from the ESS Copy Services tasks environment to the DS Copy Services environment. The Copy Services interfaces described are the ESS graphical user interface (GUI) and the command-line interface (CLI). The Copy Services interfaces that are run directly by z/OS are not included.

This appendix includes the following sections:

- Migrating ESS Copy Services tasks to DS8000 CLI
- Copy Services commands
Migrating ESS Copy Services tasks to DS8000 CLI

All of the Advanced Copy Services functions available on the ESS are available on the DS8000. However, there are differences that need to be considered when replacing the ESS CLI with the DS8000 CLI:

- Point-in-Time Copy (FlashCopy) does not support Consistency Groups in the DS Storage Manager.
- Fibre Channel is used for Metro Mirror, Global Copy, Global Mirror, and Metro/Global Mirror.
- The DS Storage Manager runs real time only (no saved tasks), whereas the DS8000 CLI can be started real time or with a saved script.
- The DS8000 CLI supports both the ESS 800/750 and the DS8000.
- ESS must be at Licensed Internal Code (LIC) level 2.4.3.15 or higher.

Review the ESS tasks to migrate

Review what ESS Copy Services tasks you want to migrate. The tasks can be viewed from the ESS GUI or the ESS CLI. Figure B-1 shows the saved tasks listed in the ESS GUI.

Highlight the task and click the information panel. Clicking the panel displays the task information that will be used for converting to the DS8000 CLI. The information from the task Flash10041005 is shown in Figure B-2. This task establishes a FlashCopy relationship between source volume 004-23953 and target volume 005-23953 with the No Background Copy option.

Figure B-1  ESS Copy Services GUI tasks panel

Figure B-2  ESS task information
The same task list can be generated using the ESS CLI command `esscli list task` as shown in Figure B-3. This output shows the same tasks as shown with the GUI.

```
   esscli list task --s copy_services-server --u csadmin --p passw0rd
```

<table>
<thead>
<tr>
<th>Task Name</th>
<th>Type</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>H_Epath_test16</td>
<td>PPRCEstablishPaths</td>
<td>NotRunning</td>
</tr>
<tr>
<td>H_Epath_test17</td>
<td>PPRCEstablishPaths</td>
<td>NotRunning</td>
</tr>
<tr>
<td>Brocade_pr_lss10</td>
<td>PPRCEstablishPair</td>
<td>NotRunning</td>
</tr>
<tr>
<td>Brocade_pr_lss11</td>
<td>PPRCEstablishPair</td>
<td>NotRunning</td>
</tr>
<tr>
<td>Flash10041005</td>
<td>FCEstablish</td>
<td>NotRunning</td>
</tr>
</tbody>
</table>

*Figure B-3*  The `esscli list task`

Select Flash10041005 and use the ESS CLI command `esscli show task` to look at the details of the task (Figure B-4). In this output, the source and target volumes are listed as 1004 and 1005. This LUN format is used in the DS8000 CLI.

```
   esscli show task --s copy_services_server --u csadmin --p passw0rd --d "name=Flash10041005"
```

```
   Taskname=Flash10041005
   Tasktype=FCEstablish
   Options=NoBackgroundCopy
   SourceServer=2105.23953
   TargetServer=2105.23953
   SourceVol TargetVol
   1004     1005
```

*Figure B-4*  The `esscli show task`

Review the specific server scripts that perform tasks that set up and start saved ESS CLI tasks. These server scripts might have to be edited or translated to run the equivalent DS8000 CLI commands.
Another example of a saved task list with the ESS GUI is displayed in Figure B-5. In this example, there are four saved tasks to establish and remove PPRC paths and pairs.

![Figure B-5 PPRC task list](image)

In the task to establish PPRC paths named `tstpths`, there is a single path in the list from source 21968:00 to 21968:01 (Figure B-6).

![Figure B-6 Establish PPRC paths task](image)
Look at the task named est1011 to establish the PPRC pairs in the information panel as shown in Figure B-7. This example is establishing Extended Distance pairs (now known as Global Copy). The information panel displays 11 volumes in source and target columns.

![Information Panel]

Figure B-7 Establish PPRC pairs task
The next saved task in the list is named del01. This task terminates the pairs created in est1011 (Figure B-8).

![Information Panel](image)

Figure B-8 Terminate PPRC pairs information window

**Important:** Open systems volume IDs on the ESS are given in an 8-digit format: xxx-ssss, where xxx is the LUN ID and ssss is the serial number of the ESS. The fixed block LUN ID used for DS8000 CLI must be the xxx number in the ESS format plus 1000. If you start the flash copy command on the ESS with the DS8000 CLI, the fixed block LUNs must be specified as 1xxx to avoid overwriting the following CKD volumes:

- 0000 to OFFF: System z CKD volumes (4096 possible addresses)
- 1000 to 1FFF: Open systems fixed block LUNs (4096 possible addresses)

**Convert the individual tasks**

This section list the ESS CLI tasks that will be translated to equivalent DS8000 CLI commands. These commands can be saved as scripts to be run in the DS8000 CLI environment. The commands can also be run as real-time commands using DS8000 CLI in interactive mode.

**Important:** Unlike the ESS GUI, the DS8000 DS Storage Manager does not save Copy Services tasks. The DS Storage Manager is only used in real-time mode for Copy Services functions.
Table B-1 lists the conversions of task parameters from the ESS CLI format to the DS8000 CLI format for the FlashCopy task reviewed in Figure B-4 on page 523.

<table>
<thead>
<tr>
<th>Task parameter</th>
<th>ESS CLI parameter</th>
<th>DS8000 CLI conversion</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tasktype</td>
<td>FCEstablish</td>
<td>mkflash</td>
<td>Establish FlashCopy</td>
</tr>
<tr>
<td>Options</td>
<td>NoBackgroundCopy</td>
<td>-nocp</td>
<td>No background copy when establishing the FlashCopy relationship</td>
</tr>
<tr>
<td>SourceServer</td>
<td>2105.23953</td>
<td>-dev IBM.2105-23953</td>
<td>Storage unit device ID</td>
</tr>
<tr>
<td>TargetServer</td>
<td>2105.23953</td>
<td>N/A</td>
<td>DS8000 CLI does not use a target server for FlashCopy commands</td>
</tr>
<tr>
<td>Source and Target volumes</td>
<td>1004 1005</td>
<td>1004:1005</td>
<td>Source volume separated by a colon from the target volume in DS8000 CLI</td>
</tr>
</tbody>
</table>

Table B-2 lists the conversions of ESS Copy Services parameters from the DS8000 CLI format for the PPRC task est1011. This table focuses on establishing PPRC as related to the saved task rather than all the saved tasks shown previously. The remaining tasks are converted to DS8000 CLI in the next section.

<table>
<thead>
<tr>
<th>Task parameter</th>
<th>ESS Copy Services parameters</th>
<th>DS8000 CLI conversion</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tasktype</td>
<td>Establish PPRC Extended Distance</td>
<td>mkpprc</td>
<td>Establish extended distance PPRC (Global Copy)</td>
</tr>
<tr>
<td>Options</td>
<td>Do Not Copy Vol Secondary Online OK PPRC Extended Distance</td>
<td>-nocp -tgtonline -type gcp</td>
<td>Establish pairs with no background copy and target volume online as Global Copy</td>
</tr>
<tr>
<td>SourceServer</td>
<td>2105-21968</td>
<td>-dev IBM.2105-21968</td>
<td>Source storage unit device ID</td>
</tr>
<tr>
<td>TargetServer</td>
<td>2105-21968</td>
<td>-remotedev IBM.2105-21968</td>
<td>Remote storage unit device ID</td>
</tr>
<tr>
<td>Source and Target volumes</td>
<td>0080 1080 ...</td>
<td>0080-008B:1080-108B</td>
<td>Source and target volume ranges separated by a (;) colon in DS8000 CLI</td>
</tr>
</tbody>
</table>
Use the information in Table B-1 on page 527 to convert the ESS CLI commands to establish FlashCopy into DS8000 CLI commands. Issue the `mkflash` command using the No Background Copy option `-nocp` to the ESS volumes 1004:1005 (Figure B-9). The relationship is queried using the DS8000 CLI command `lsflash` for the specific pair.

```
   dscli> mkflash -nocp -dev IBM.2105-23953 1004:1005
   CMUC00137I mkflash: FlashCopy pair 1004:1005 successfully created.
   dscli> lsflash -dev IBM.2105-23953 1004:1005
   ID SrcLSS SequenceNum Timeout ActiveCopy Recoding Persistent Revertible
   1004:1005 10 0 120 Disabled Disabled Disabled Disabled
```

*Figure B-9  DS8000 CLI command to establish and list a FlashCopy pair*

To create the PPRC paths seen in Figure B-6 on page 524, use the DS8000 CLI command `mkpprcpath` (Figure B-10). This example shows a single path between two ESS storage units using source LSS 00 and target LSS 01. This output is slightly different than in the information panel for `tstpths` because that task is creating paths within the same ESS.

```
   dscli> mkpprcpath -dev IBM.2105-23953 -remotedev IBM.2105-21968 -remotewwn 5005076300C09368 -srclss 00 -tgtlss 01 1004:1005
```

*Figure B-10  Creating PPRC paths on ESS with DS8000 CLI*

To create the PPRC pairs seen in Figure B-7 on page 525, use the DS8000 CLI command `mkpprc` as shown in Figure B-11. Use the following options:

- The `-type gcp` option to create the extended distance PPRC relationship
- The `-mode nocp` option to specify the Do Not Copy Vol parameter
- The `-tgtonline` option to specify the Secondary Online OK parameter

Select the source volume range 0080 to 008B paired with target volumes 0180 to 018B, as shown in the ESS task est1011.

```
   dscli> mkpprc -dev IBM.2105-23953 -remotedev IBM.2105-21968 -type gcp -mode nocp -tgtonline 0080-008B:0180-018B
```

*Figure B-11  Create PPRC pairs on ESS with DS8000 CLI*

The final task is converting the task to terminate PPRC, which is displayed in Figure B-8 on page 526, using the DS8000 CLI command `rmpprc`. This conversion is shown in Figure B-12. The saved task on the ESS did not include any options for the termination. Therefore, the new DS8000 CLI command does not include any of the available options for this command.

```
   dscli> rmpprc -dev IBM.2105-23953 -remotedev IBM.2105-21968 0080-008B:0180-018B
```

*Figure B-12  Terminate PPRC pairs on ESS with DS8000 CLI*
Copy Services commands

The Flash Copy and RMC Copy Services DS8000 CLI commands and the equivalent ESS Copy Services commands are listed in Table B-3. The commands are grouped by type of function. The ESS CLI can start saved tasks created through the ESS GUI by issuing the command `rsExecuteTask`. The functions that are executed in a saved task have a corresponding DS8000 CLI command. This table is the correlation between the functions started in a saved task and the DS8000 CLI command.

Table B-3  DS8000 CLI copy services commands and equivalent ESS CLI commands

<table>
<thead>
<tr>
<th>DS8000 CLI command</th>
<th>ESS Copy Services</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Flash Copy Commands</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>mkflash</code></td>
<td>FCEstablish</td>
<td>Establishes FlashCopy</td>
</tr>
<tr>
<td><code>lsflash</code></td>
<td>rsFlashCopyQuery</td>
<td>Lists FlashCopy relationship</td>
</tr>
<tr>
<td><code>mkremoteflash</code></td>
<td>FCEstablish with Inband option</td>
<td>Creates FlashCopy pair on remote system through RMC conduit</td>
</tr>
<tr>
<td><code>resyncflash</code></td>
<td>FCEstablish with Increment and Persistent options</td>
<td>Updates a point in time copy, copying only the tracks that have been updated since the last copy</td>
</tr>
<tr>
<td><code>resyncremoteflash</code></td>
<td>FCEstablish with Inband, Increment, and Persistent options</td>
<td>Same as resyncflash except on a remote system issued through an RMC conduit</td>
</tr>
<tr>
<td><code>reverseflash</code></td>
<td>FCEstablish with Reverse Restore option</td>
<td>Reverses the FlashCopy direction so that the source becomes the target</td>
</tr>
<tr>
<td><code>revertremoteflash</code></td>
<td>FCEstablish with Inband and Revertible options</td>
<td>Restores data on the source volume to its most recent consistency formation, issued through RMC conduit</td>
</tr>
<tr>
<td><code>rmflash</code></td>
<td>FCWithdraw</td>
<td>Removes the FlashCopy relationship</td>
</tr>
<tr>
<td><code>rmremoteflash</code></td>
<td>FCWithdraw with Inband option</td>
<td>Removes the FlashCopy relationship on the remote storage unit through RMC conduit</td>
</tr>
<tr>
<td><code>unfreezeflash</code></td>
<td>Flash Copy create consistency group</td>
<td>Resets a FlashCopy consistency group previously established with the -freeze parameter</td>
</tr>
<tr>
<td><strong>PPRC Commands</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>fallbackpprc</code></td>
<td>PPRCEstablishPair with PPRC Failback option</td>
<td>Copies data from the source volume to the target volume to resume mirroring</td>
</tr>
<tr>
<td><code>failoverpprc</code></td>
<td>PPRCEstablishPair with PPRC Failover option</td>
<td>Site switch causing the target volumes to become the source volumes</td>
</tr>
</tbody>
</table>
Consider the following notes about Copy Services:

- A 2107 ESCON I/O port is only used for System z host attachment. It cannot be configured as an RMC path.

- A 2107 Fibre Channel port is configured for either SCSI-FCP or FICON protocol. Like the 2105, a FICON port is restricted to the point-to-point/switched fabric topology setting. A FICON I/O port is used for System z host attachment and cannot be configured as an RMC path.

- A 2107 Fibre Channel SCSI-FCP I/O port can be configured for the point-to-point (switched) fabric of FC-AL connection topologies. A port that uses point-to-point (switched) fabric topology can be used for Open systems host I/O and RMC path configurations simultaneously.

- A 2107 storage facility image can contain up to 65,280 volumes. A 2105 storage unit can contain up to 4096 FB volumes and 4096 CKD volumes.

---

<table>
<thead>
<tr>
<th>DS8000 CLI command</th>
<th>ESS Copy Services</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>freezepprc</td>
<td>PPRCFreezeGroup</td>
<td>Creates new RMC consistency group and removes RMC paths</td>
</tr>
<tr>
<td>lsavailpprcport</td>
<td>n/a</td>
<td>Lists ports available for RMC paths between two storage units</td>
</tr>
<tr>
<td>lspprc</td>
<td>n/a</td>
<td>Displays the RMC status information</td>
</tr>
<tr>
<td>lspprcpath</td>
<td>list pprcpaths</td>
<td>Displays RMC paths for the specified source and target LSSs</td>
</tr>
<tr>
<td>mkpprc</td>
<td>PPRCEstablishPair</td>
<td>Establishes RMC pairs</td>
</tr>
<tr>
<td>mkpprcpath</td>
<td>PPRCEstablishPaths</td>
<td>Establishes RMC paths</td>
</tr>
<tr>
<td>pausepprc</td>
<td>PPRCSuspendPair</td>
<td>Pauses an existing RMC relationship</td>
</tr>
<tr>
<td>resumepprc</td>
<td>n/a</td>
<td>Resumes RMC relationship</td>
</tr>
<tr>
<td>rmpprc</td>
<td>PPRCTerminatePair</td>
<td>Removes RMC relationship</td>
</tr>
<tr>
<td>rmpprcpath</td>
<td>PPRCRemovePaths</td>
<td>Removes RMC paths</td>
</tr>
</tbody>
</table>
Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

For information about ordering these publications, see “Help from IBM” on page 531. Note that some of the documents referenced here may be available in softcopy only.

- *DS8000 Copy Services for IBM System z*, SG24-6787
- *IBM System Storage DS8000: Copy Services in Open Environments*, SG24-6788
- *Implementing the IBM System Storage SAN Volume Controller V5.1*, SG24-6423
- *Implementing the IBM System Storage SAN Volume Controller V6.1*, SG24-7933

You can search for, view, download or order these documents and other Redbooks, Redpapers, Web Docs, draft and additional materials, at the following website:

ibm.com/redbooks

Other publications

These publications are also relevant as further information sources:

- *DS8000 Copy Services for IBM System z*, SG24-6787
- *z/OS DFSMS Advanced Copy Services*, SC35-0428

Online resources

These websites are also relevant as further information sources:

- Data Migration Service
  http://www.ibm.com/servers/storage/datamobility/
- IBM System Storage

Help from IBM

IBM Support and downloads

ibm.com/support
Index

A
access method services (AMS) 450
ACF2 448
activation phase 401
Active in Copy 422
active/active 135, 137, 143, 168, 173
active/passive 135, 137, 139, 168, 173
Add Mirror 279
agent system 396, 485
responsibilities 398, 487
AIX LVM 330
alias 436
ALLOWCM 419
ALLOWmirrorchange 400
ALTER authority 399
alternate couple data set 424
alternate path 327
AMS see access method services (AMS)
authentication 51
auxiliary SVC cluster 246
auxiliary volume 183

B
back-end application 178
background copy 135–136, 148, 158
backup 10
bandwidth 44
basic catalog structure (BCS) 440, 450
basic disk 268
BCS see basic catalog structure (BCS)
blockdev 382

C
cache 43, 91, 162, 431
destage 186
catalog structure 450
CCHH 425
cell type 451
CESTPAIR 71, 75
cfgmnr 90, 334, 343, 357
CFRM see coupling facility resource management (CFRM)
channel 425
extension 55
CHECK TARGET 421
checkpoint 434
chfs 92
chunks 185–186
CICS 444
CKD 49
CLASS=DATASET 405
CLI see command-line interface (CLI)
cluster 148–150
CNTL 412
command-line interface (CLI) 47
COMMDS see communications data set (COMMDS)
communication links 454
communications data set (COMMDS) 396, 398, 401,
403, 412, 419, 485–487
completion phase 435
concurrent instances 199
concurrent volumes 429
configuration node 178
consistency groups 182, 522
consolidation 266
Copy Pending 87
copy phase 401, 435, 446
Copy Services tasks 526
Couple data sets 424
coupling facility resource management (CFRM) 456
policy 456
CQUERY 74
cquery 100–101
create pairs 71–72
create paths 67
cross-system coupling facility (XCF) 454, 456
CS Domain 54

D
DASDVOL 405
Data Facility Storage Management Subsystem (DFSMS) 437
data integrity 398, 420, 458, 486
data migration 133–135, 142
back-out 148, 169–170
checklist 170
monitoring 158
object 146–148, 157
offline 185
online 188
steps 138, 150, 169
synchronization 151
target 135
volume 146
data redirection 437
Data Set Services (DFSMSdss) 438
datapath 277
dataset level migration 434
DB2 444
dd 9–10
deadlock 447
deadly embrace 458
delete pairs 72
dense wave division multiplexing (DWDM) 43
dependent writes 182
Destination Name 146, 157
Destination Pool 146, 157
device geometry 12
Device Manager 161
device mapper 376
device special file 322, 329
df command 92, 292, 299, 308
DFDSS 444
DFSMdss see Data Set Services (DFSMdss)
DFSMS see Data Facility Storage Management Subsystem (DFSMS)
DFSMSdp 438
DFSMShsm see Hierarchical Storage Manager (DFSMSHsm)
DIAGNOSE 451
disk group 319
Disk Magic 46
diskpart 164
disruptive 297
diversion 447, 457
Diversion phase 435, 457
divert 447
dm_list 150, 153
DS CLI 49, 522
DS Storage Manager 47
DS4000 138, 144
DS6000 137
DS8000 137
  firmware 267
  serial number 334
dual cluster Metro Mirroring 188
DWDM see dense wave division multiplexing (DWDM)
dynamic disk 268, 272
Dynamic Multipathing (DMP) 307
dynamic swap 434
dynamic takeover 396

E
ENQ 458
environment variables 155–156
ESCON 49, 56, 425
ESS 137, 144, 157, 330, 350
  CLI 522
  GUI 522
  port 322
ESS Specialist 52
ESTPAIR 72
event log 160
EXAMINE 453
Extend Volume 286
Extended Distance 525
Extended Mirrored phase 441
extent 179, 181, 186

F
fail-over 137, 168
FastCopy 422, 425
FCEstablish 527
fdisk 381
FICON 49, 425
FlashCopy 176
freezeprpc 60
front-end application 178
fsck 92
Full Duplex 72, 87
Full Speed Copy 422, 425, 429
fuzzy copy 45

G
GDG see generation data group (GDG)
generation data group (GDG) 420, 439
Global Copy performance 46
global enqueue 458
Global Mirror 42, 185
global resource serialization (GRS) 454, 457
go to sync 45, 73
d graphical user interface (GUI) 47
GRS see global resource serialization (GRS)
GUI see graphical user interface (GUI)

H
Hardware Management Console (HMC) 47
hardware reserve 456–457
HBA 267, 289, 305
  firmware level 267
hdisk 337
Hierarchical Storage Manager (DFSMSHsm) 438
History option 399, 404
HMC see Hardware Management Console (HMC)
host attachment procedures 150
host computer 176
host server 134–135, 150
  I/O requests 135
HP-UX Volume Manager 320

I
I/O enclosure 55
I/O group 178
IBM XIV
data migration solution 134
  storage 135
ICF catalog 450
ICKDSF 47, 50, 72, 425, 443
IDCAMS 436, 449
idling 259
IEBCOPY 403
image mode MDisk 194
image mode VDisk 185, 196
importvg 90–91
Inhibit Cache Load 431
initialization phase 400–401
initialization rate 159
initiator 134, 137, 139
Input/Output Supervisor (IOS) 430
insf 322
Intercluster Metro Mirror 184
intercluster Metro Mirror 184, 188
Intracluster Metro Mirror 184
intracluster Metro Mirror  184, 188
IOCS  99
IOS see Input/Output Supervisor (IOS)
ioscan  322
ISMF  437

J
JCL  444
JFS2  91–92

K
Keep Source Updated  136, 147, 158
key-sequenced data set (KSDS)  450
KSDS see key-sequenced data set (KSDS)

L
label  95, 310
last extent  195
lavailpprcport  63
LCU see logical control unit (LCU)
LDM see Windows Logical Disk Manager (LDM)
license key  404, 418, 488
link  44
Linux x86  140
listing utility  439
Local Page  423
logical control unit (LCU)  422
Logical Disk Manager (LDM)  266
Logical Volume Management 2 (LVM2)  376
Logical Volume Manager (LVM)  266
logredo  92
LPAR  83, 396, 408, 420, 485
lsfbvol  83
lsv  344
lspprc  72, 87–88
lspprcpath  85
lsrcrelationship  256
lss  83
lsvg  333, 335, 344, 360
lsvpfg  334, 347, 381
LUN ID  135, 145–146, 150, 334
LUN mapping  145
LUN numbering  140, 173
LUN0  140, 144, 171, 173
lextend  328
LVM see Logical Volume Manager (LVM)
LVM2 see Logical Volume Management 2 (LVM2)
lvreduce  328

M
managed disk  179
managed disk group  179
managed mode MDisk  194
management console  177
Master
Console  178
responsibilities  487
SVC cluster  246
VDisk  251
master system  396, 485
responsibilities  398, 487
master volume  183
MAX CHANNEL_IO  446
MAX DEVICE_IO  446
max_initialization_rate  159–161
max_resync_rate  159
max_syncjob_rate  159
MAXIO  446
MAXTRK  446
MDG  179
MDisk  179
modes  194
member name  441
metainit  290
metastat  299–300, 303
metattach  290, 300
Metro Mirror  43, 176, 183, 188
link  44
monitor  254
performance  44
Microsoft Cluster Server (MSCS)  148
migrate a VDisk  185, 196
migrate between MDGs  185, 196
migrate data  194
migratepv  329, 331, 334
migratepv -p  336
migration
algorithm  185
group  397, 450, 485
pair  397, 485
progress  198
session  397, 485
speed  137, 158
tips  195, 200
Mirror phase  435
Mirror Synchronization phase  446
mirroring  265
process  278
mkfbvol  82
mkflash  528
mkiv  344
mkvcopy  329, 334, 339, 345
mkpprc  73
mkpprcpath  528
mkrcconsistgrp  252
MPIO  267
MSCS see Microsoft Cluster Server (MSCS)
Multi-Image Manager  457
Multipath Subsystem Device Driver (SDD)  182, 333, 377, 380
multipathing  173
MVS Write-to-Operator/Write-to-Operator with Reply (WTO/WTOR)  400

N
naming convention  425
naming standard  436
no source updating  135–136
option 135
source volume 396, 485
special migration 195
split mirroring 265
SSMzOS 452
stack 186
stale partition 346
star complex 456
started task 396
startrcconsistgr 255
startrcrelationship 256
stmsboot 290, 292
Storage Complex 54
storage group 438
storage subsystem 176
storage system 133–134
storage virtualization 376
submirror 288, 297, 299
SuSE 140
suspend pairs 72
suspended 71
SVC see SAN Volume Controller (SVC)
svcinfo lsfreeextents 198
svcinfo lsmdiskextent 198
svcinfo lsmigrate 198
svcinfo lsVdiskextent 198
svctask mkpartnership 249
SVM see Solaris Volume Manager (SVM)
swap migration 397, 405, 485
synchronization 151, 159, 397, 485
Synchronization phase 397, 401, 416, 435, 485
synchronized 135, 154, 159
synchronous reads 186
synchronous writes 186
syncvg 329, 339, 345
syslog 417
SYSSOPTN 404, 418, 420
sysplex 455–456
couple data set 456
sysplex failure management (SFM) 456
policy 456
System Authorization Facility (SAF) 399, 405
System Display and Search Facility (SDSF) 432
System Page Pool 423
System z 99
interface 49

T

target 135, 137, 144
target volume 396, 485
target_list 152, 159
TCO see total cost of ownership (TCO)
TDMEXEC 410
TDMF see Transparent Data Migration Facility (TDMF) for z/OS
Termination phase 401, 421
Test Data Migration 148–149, 154, 158
thick to thin migration 162
thin provisioning 162
Tivoli Storage Productivity Center (TPC) for Replication

47
total cost of ownership (TCO) 434
TPC see Tivoli Storage Productivity Center (TPC) for Replication
track geometry 396, 485
transfer rate 159
transitions 194
Transparent Data Migration Facility (TDMF) for z/OS 395–396, 483–484
agent system 396, 485
responsibilities 398, 487
architecture 397, 486
best practices 418, 488
hardware 401, 487
initialization 399
license key 404
master system 396, 485
responsibilities 398, 487
migration group 397, 485
migration pair 397, 485
migration session 397, 485
process flow 398
source volume 396, 485
swap migration 397, 485
system initialization 399
target volume 396, 485
TSO Monitor 431

TSO 47, 103

U

umount command 92
unfreezepprc 60
Unidentified Systems 420
uninterruptible power supply (UPS) 178
unmanaged MDisk 194
UPDATE authority 399
UPS see uninterruptible power supply (UPS)

V

VCMDB see Volume Configuration Management Database LUNs (VCMDB)
VDisks 176, 179
image mode migration concept 194
migration 181
vendor identification (VID) 291
Veritas Storage Foundation 304
Veritas Volume Manager (VxVM) 92, 266, 304–305
vgdisplay 324
vgextend 325
vgmodify 326
vgreduce 328, 386
vgscan 382
VID see vendor identification (VID)
virtual disks 179
Virtual Storage Access Method (VSAM) 440
volser 439
Volume Configuration Management Database LUNs (VCMDB) 294, 307
volume table of contents (VTOC) 450, 458
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>volumes</td>
<td></td>
</tr>
<tr>
<td>acknowledgement</td>
<td>400</td>
</tr>
<tr>
<td>confirmation</td>
<td>400</td>
</tr>
<tr>
<td>I/O redirect phase</td>
<td>401</td>
</tr>
<tr>
<td>initialization</td>
<td>401</td>
</tr>
<tr>
<td>pair assignment</td>
<td>66</td>
</tr>
<tr>
<td>selection</td>
<td>401</td>
</tr>
<tr>
<td>vpath</td>
<td>342, 347, 381–382</td>
</tr>
<tr>
<td>VPSMS</td>
<td>444</td>
</tr>
<tr>
<td>VSAM see</td>
<td></td>
</tr>
<tr>
<td>Virtual Storage Access Method (VSAM)</td>
<td></td>
</tr>
<tr>
<td>VSAM volume data set (VVDS)</td>
<td>450, 458</td>
</tr>
<tr>
<td>VSAM volume record (VVR)</td>
<td>451</td>
</tr>
<tr>
<td>VTOC see</td>
<td></td>
</tr>
<tr>
<td>volume table of contents (VTOC)</td>
<td></td>
</tr>
<tr>
<td>VTOCIX</td>
<td>440</td>
</tr>
<tr>
<td>VSAM volume data set (VVDS)</td>
<td></td>
</tr>
<tr>
<td>VVR see</td>
<td></td>
</tr>
<tr>
<td>VSAM volume record (VVR)</td>
<td></td>
</tr>
<tr>
<td>vxdg</td>
<td>98, 306</td>
</tr>
<tr>
<td>vxdisk list</td>
<td>310</td>
</tr>
<tr>
<td>vxdiskadm</td>
<td>306, 312, 314</td>
</tr>
<tr>
<td>vxplex</td>
<td>306, 318</td>
</tr>
<tr>
<td>vxprint</td>
<td>308, 319</td>
</tr>
<tr>
<td>vxtask</td>
<td>317</td>
</tr>
<tr>
<td>VxVM see</td>
<td></td>
</tr>
<tr>
<td>Veritas Volume Manager (VxVM)</td>
<td></td>
</tr>
<tr>
<td>W</td>
<td></td>
</tr>
<tr>
<td>Windows Logical Disk Manager (LDM)</td>
<td>268</td>
</tr>
<tr>
<td>worldwide port name (WWPN)</td>
<td>59, 140, 142, 144, 153, 166</td>
</tr>
<tr>
<td>write access</td>
<td>258</td>
</tr>
<tr>
<td>WTO/WTOR see</td>
<td></td>
</tr>
<tr>
<td>MVS Write-to-Operator/ Write-to-Operator with Reply (WTO/WTOR)</td>
<td></td>
</tr>
<tr>
<td>WWPN see</td>
<td>worldwide port name (WWPN)</td>
</tr>
<tr>
<td>X</td>
<td></td>
</tr>
<tr>
<td>XCF see</td>
<td></td>
</tr>
<tr>
<td>cross-system coupling facility (XCF)</td>
<td></td>
</tr>
<tr>
<td>XCLI see</td>
<td></td>
</tr>
<tr>
<td>XIV command-line interface (XCLI)</td>
<td>152</td>
</tr>
<tr>
<td>XIV Storage System</td>
<td>133–134</td>
</tr>
<tr>
<td>Z</td>
<td></td>
</tr>
<tr>
<td>z/OS Dataset Mobility Facility (zDMF)</td>
<td>433</td>
</tr>
<tr>
<td>parameters</td>
<td>446</td>
</tr>
<tr>
<td>performance</td>
<td>445</td>
</tr>
<tr>
<td>storage requirements</td>
<td>447</td>
</tr>
<tr>
<td>zDMF see</td>
<td>z/OS Dataset Mobility Facility (zDMF)</td>
</tr>
<tr>
<td>zone</td>
<td>288, 338, 379</td>
</tr>
<tr>
<td>zoning</td>
<td>139–140, 149, 165</td>
</tr>
</tbody>
</table>
Data migration has become a mandatory and regular activity for most data centers. Companies need to migrate data not only when technology needs to be replaced, but also for consolidation, load balancing, and disaster recovery.

This IBM Redbooks publication addresses the aspects of data migration efforts while focusing on the IBM System Storage as the target system. Data migration is a critical and complex operation, and this book provides the phases and steps to ensure a smooth migration. Topics range from planning and preparation to execution and validation.

The book also reviews products and describes available IBM data migration services offerings. It explains, from a generic standpoint, the appliance-based, storage-based, and host-based techniques that can be used to accomplish the migration. Each method is explained including the use of the various products and techniques with different migration scenarios and various operating system platforms.

This document targets storage administrators, storage network administrators, system designers, architects, and IT professionals who design, administer or plan data migrations in large data Centers. The aim is to ensure that you are aware of the current thinking, methods, tools, and products that IBM can make available to you. These items are provided to ensure a data migration process that is as efficient and problem-free as possible.

The material presented in this book was developed with versions of the referenced products as of June 2011.