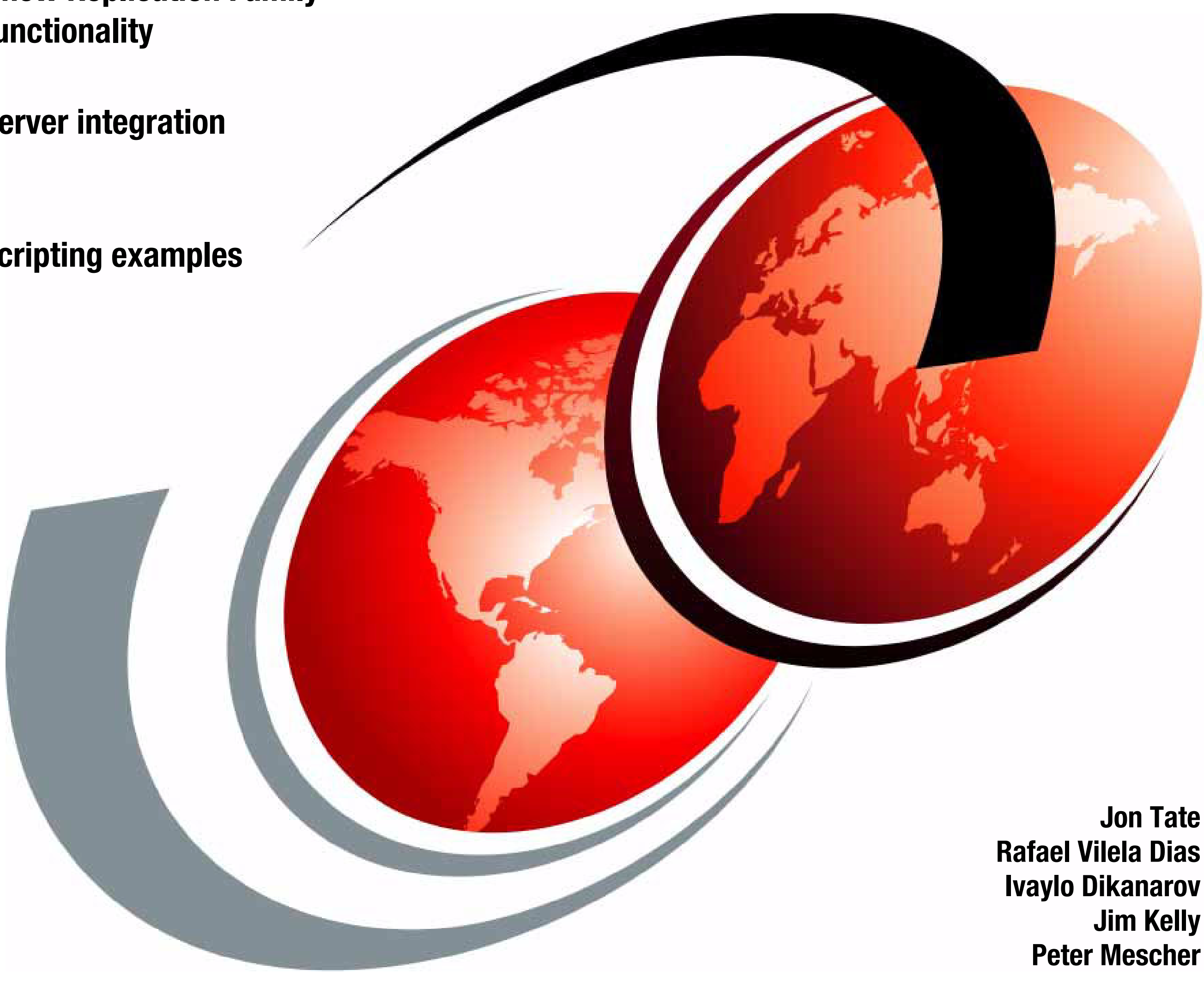
IBM System Storage SAN Volume Controller and Storwize V7000 Replication Family Services

Describes new Replication Family Services functionality

Provides server integration examples

Includes scripting examples



Jon Tate
Rafael Vilela Dias
Ivaylo Dikanarov
Jim Kelly
Peter Mescher

Redbooks

ibm.com/redbooks

International Technical Support Organization

IBM System Storage SAN Volume Controller and Storwize V7000 Replication Family Services

March 2013

SG24-7574-02

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

Third Edition (March 2013)

This edition applies to IBM System Storage SAN Volume Controller and IBM System Storage Storwize V7000 Replication Family Services at an SVC Version6.4 code level.

Contents

Notices xi
Trademarks xii

Summary of changes xiii
March 2013, Third 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

Chapter 1. Introduction to Replication Family Services 1
1.1 FlashCopy 2
1.2 Metro Mirror and Global Mirror 2
1.2.1 Global Mirror with Change Volumes 2
1.3 Volume Mirroring function 2
1.3.1 Stretched Cluster Volume Mirroring 3

Chapter 2. Planning for Replication Family Services 5
2.1 High-level design 6
2.1.1 Defining your project 6
2.1.2 Functional requirements 6
2.1.3 Non-functional requirements 7
2.1.4 Prior to implementation 7
2.2 Business continuity and productivity goals 8
2.2.1 Data protection 8
2.2.2 Storage efficiency 8
2.3 High availability 9
2.3.1 Volume Mirroring for high availability 9
2.3.2 SVC Stretched Cluster for high availability 9
2.4 Backup and restore 10
2.4.1 FlashCopy for backup and restore 10
2.4.2 Using Reverse FlashCopy to restore a volume 11
2.4.3 Backup and restore references 12
2.5 Disaster recovery 12
2.5.1 Disaster recovery site location 13
2.5.2 Metro Mirror for disaster recovery 14
2.5.3 Global Mirror for disaster recovery 14
2.5.4 FlashCopy in a Metro or Global Mirror recovery 14
2.5.5 Global Mirror with Change Volumes for disaster recovery 15
2.5.6 SVC Stretched Cluster for disaster recovery 16
2.5.7 Three-site replication for disaster recovery 17
2.5.8 Further reading for disaster recovery 17
2.6 Creating volume copies 17
2.6.1 Single Volume Mirroring 17
2.6.2 FlashCopy (volumes and Consistency Groups) 18
2.6.3 Metro or Global Mirror 19
2.7 Moving volumes 19

2.7.1 migratevdisk . . . 19
2.7.2 Volume Mirroring usage . . . 19
2.7.3 movevdisk . . . 19
2.7.4 Metro or Global Mirror . . . 20
2.8 Capacity planning . . . 20
2.8.1 Performance monitoring on SVC and Storwize V7000 . . . 21
2.8.2 Using Disk Magic to predict the impact of Metro and Global Mirror . . . 23
2.8.3 FlashCopy considerations for disk performance . . . 26
2.8.4 Grain size . . . 26
2.9 Planning for supportability . . . 27
2.9.1 Understanding consistency . . . 27
2.9.2 Planning for host attachments . . . 30
2.9.3 Extended Quorum Support for SVC Stretched Cluster . . . 32
2.9.4 Support Notifications . . . 33

Chapter 3. Metro Mirror and Global Mirror . . . 35
3.1 Terminology . . . 36
3.2 Metro and Global Mirror overview . . . 36
3.3 Partnerships between systems . . . 37
3.3.1 Layer concept . . . 39
3.3.2 Partnership topologies . . . 40
3.3.3 Partnership heartbeat . . . 42
3.4 Network connectivity for Metro and Global Mirror . . . 42
3.4.1 Fibre Channel over Ethernet (FCoE) . . . 43
3.4.2 Planning your network for Metro or Global Mirror . . . 45
3.5 Relationships between volumes or Consistency Groups . . . 46
3.5.1 Consistency Groups . . . 47
3.6 Initial synchronization . . . 48
3.6.1 Default synchronization across the network . . . 48
3.6.2 Tape-based synchronization . . . 48
3.7 Relationship states . . . 50
3.7.1 State definitions . . . 50
3.7.2 State descriptions . . . 51
3.7.3 Stop, Start, and Switch commands . . . 54
3.8 Tunable parameters and limits . . . 56
3.8.1 I/O Group memory allocation for Metro and Global Mirror . . . 56
3.8.2 Partnership bandwidth (total background copy limit) . . . 57
3.8.3 relationshipbandwidthlimit . . . 58
3.8.4 rcbuffersize . . . 59
3.8.5 gmlinktolerance and gmmaxhostdelay . . . 59
3.8.6 Configuration limits . . . 59
3.9 Metro Mirror . . . 60
3.10 Global Mirror . . . 61
3.10.1 gmlinktolerance and gmmaxhostdelay . . . 62
3.10.2 rcbuffersize . . . 63
3.10.3 Link delay simulation . . . 64
3.11 Global Mirror with Change Volumes . . . 64
3.11.1 Recovery point objective using Change Volumes . . . 67
3.11.2 Bandwidth sizing example for Change Volumes . . . 68
3.11.3 Disk sizing for Change Volumes . . . 69
3.11.4 Migrating from Global Mirror to Global Mirror with Change Volumes . . . 70
3.12 Intra-cluster replication . . . 71
3.13 Three-site replication with SVC Stretched Cluster and Metro Mirror or Global Mirror. 72

Chapter 4. Implementing Metro Mirror and Global Mirror 75
4.1 Metro Mirror using the CLI 76
4.1.1 Setting up Metro Mirror 77
4.1.2 Creating a partnership - Metro Mirror 77
4.1.3 Creating Metro Mirror relationships. 81
4.1.4 Starting Metro Mirror 83
4.1.5 Monitoring the background copy progress 84
4.1.6 Stopping and restarting Metro Mirror 86
4.1.7 Changing the copy direction on Metro Mirror 89
4.1.8 Creating multiple partnerships 92
4.2 Global Mirror using the CLI 98
4.2.1 Setting up Global Mirror 99
4.2.2 Creating a partnership - Global Mirror. 100
4.2.3 Changing link tolerance and system delay simulation 101
4.2.4 Creating Global Mirror relationships 103
4.2.5 Starting Global Mirror 105
4.2.6 Monitoring the background copy progress 107
4.2.7 Stopping and restarting Global Mirror 108
4.2.8 Switching the direction on Global Mirror 112
4.3 Global Mirror with Change Volumes using the CLI 114
4.3.1 Change Volume requirements 114
4.3.2 Changing Global Mirror to Global Mirror with Change Volumes 115
4.3.3 Creating the thin-provisioned Change Volumes 115
4.3.4 Stopping the stand-alone remote copy relationship 116
4.3.5 Setting cyclingmode on the stand-alone remote copy relationship 116
4.3.6 Setting the Change Volume on the master volume. 116
4.3.7 Setting the Change Volume on the auxiliary volume 117
4.3.8 Starting the stand-alone relationship with cyclingmode 118
4.3.9 Stopping the Consistency Group to allow setting cyclingmode. 119
4.3.10 Setting cyclingmode on the Consistency Group 120
4.3.11 Setting the Primary Change Volumes for the Consistency Group 120
4.3.12 Setting the secondary Change Volumes for the Consistency Group 121
4.3.13 Starting the Consistency Group with Change Volumes 123
4.4 Implementation using the GUI. 124
4.4.1 Creating a partnership between two systems (clusters) 125
4.4.2 Creating stand-alone relationships between volumes. 128
4.4.3 Creating a Consistency Group 131
4.4.4 Renaming a Consistency Group 137
4.4.5 Renaming a relationship 137
4.4.6 Moving a stand-alone remote copy relationship to a Consistency Group 138
4.4.7 Removing a remote copy relationship from a Consistency Group 139
4.4.8 Starting a relationship 140
4.4.9 Starting a Consistency Group 142
4.4.10 Switching the copy direction on a relationship 143
4.4.11 Switching the copy direction on a Consistency Group 145
4.4.12 Stopping a Metro Mirror or Global Mirror relationship 146
4.4.13 Stopping a Consistency Group 147
4.4.14 Deleting stand-alone relationships 149
4.4.15 Deleting a Consistency Group 150

Chapter 5. FlashCopy 151
5.1 Introduction to FlashCopy 152
5.1.1 Life without the FlashCopy function 152

5.1.2 Life with the FlashCopy function 152
5.1.3 FlashCopy function use and example cases. 153
5.2 FlashCopy characteristics 153
5.2.1 FlashCopy mapping 154
5.2.2 Number of FlashCopy mappings in a cluster 154
5.2.3 FlashCopy Consistency Groups 155
5.2.4 Multiple Target FlashCopy 156
5.2.5 Cascaded FlashCopy 157
5.2.6 Combined Multiple Target FlashCopy and Cascaded FlashCopy 158
5.2.7 Incremental FlashCopy 158
5.2.8 Thin-provisioned FlashCopy 160
5.2.9 Background copy 161
5.3 FlashCopy internals 163
5.3.1 Indirection layer. 163
5.3.2 Summary of FlashCopy indirection layer algorithm. 164
5.3.3 Discussion of grains 164
5.3.4 FlashCopy indirection layer algorithm 165
5.3.5 Interaction and dependency between Multiple Target FlashCopy mappings . . . 167
5.3.6 bitmaps 168
5.3.7 Usable bitmap space 168
5.3.8 FlashCopy maximum configurations 169
5.3.9 Summary of FlashCopy function rules and characteristics 169
5.4 FlashCopy events and states 171
5.4.1 FlashCopy mapping states 173
5.5 Dependencies between FlashCopy mappings 175
5.5.1 Linked lists of mappings and resulting dependencies. 175
5.5.2 Stopping FlashCopy mappings within a tree. 180
5.6 Practical example of a mapping tree 182
5.6.1 FlashCopy mapping setup 182
5.6.2 Workflow to set up and run FlashCopy mappings. 183
5.6.3 Characteristics of this setup and notes about the steps 183
5.6.4 Workflow to set up and run FlashCopy mappings. 184
5.6.5 Characteristics of this setup and notes about the steps 185

Chapter 6. Implementing FlashCopy 187
6.1 Copy considerations 188
6.1.1 Command syntax and online help. 188
6.1.2 Interoperation with Metro Mirror and Global Mirror 190
6.1.3 FlashCopy: simulated scenario. 191
6.2 FlashCopy - CLI operation 192
6.2.1 Setting up FlashCopy 192
6.2.2 Creating a FlashCopy Consistency Group 193
6.2.3 Creating a FlashCopy mapping 193
6.2.4 Preparing (pre-triggering) the FlashCopy mapping. 195
6.2.5 Preparing (pre-triggering) the FlashCopy Consistency Group 196
6.2.6 Starting (triggering) FlashCopy mappings. 197
6.2.7 Starting (triggering) FlashCopy Consistency Group 198
6.2.8 Monitoring the FlashCopy progress 199
6.2.9 Stopping the FlashCopy mapping. 199
6.2.10 Stopping the FlashCopy Consistency Group 200
6.2.11 Deleting the FlashCopy mapping 201
6.2.12 Deleting the FlashCopy Consistency Group 202
6.2.13 Migrating a volume to a thin-provisioned volume 202

6.2.14 Reverse FlashCopy 207
6.2.15 Split-stopping FlashCopy maps 209
6.3 Example: FlashCopy using the CLI 209
6.3.1 Checking Volumes and the Metro Mirror relationship 210
6.3.2 Creating FlashCopy mappings 213
6.3.3 Creating a Consistency Group 216
6.3.4 Adding FlashCopy mappings to a Consistency Group 217
6.3.5 Preparing a FlashCopy mapping 218
6.3.6 Starting a Consistency Group 219
6.3.7 Modifying a FlashCopy mapping 222
6.3.8 Stopping a FlashCopy mapping and a Consistency Group 223
6.3.9 Deleting a FlashCopy mapping and a Consistency Group 225
6.4 FlashCopy - GUI operation 226
6.4.1 Introduction to the GUI 227
6.4.2 Creating a FlashCopy mapping 232
6.4.3 Creating and starting a snapshot preset with a single click 247
6.4.4 Creating and starting a clone preset with a single click 249
6.4.5 Creating and starting a backup preset with a single click 251
6.4.6 Creating a FlashCopy Consistency Group 254
6.4.7 Creating FlashCopy mappings in a Consistency Group 255
6.4.8 Showing dependent mappings 260
6.4.9 Moving a FlashCopy mapping to a Consistency Group 261
6.4.10 Removing a FlashCopy mapping from a Consistency Group 262
6.4.11 Modifying a FlashCopy mapping 263
6.4.12 Renaming a FlashCopy mapping 264
6.4.13 Renaming a Consistency Group 265
6.4.14 Deleting a FlashCopy mapping 266
6.4.15 Deleting a FlashCopy Consistency Group 267
6.4.16 Starting FlashCopy mappings 268
6.4.17 Starting a FlashCopy Consistency Group 269
6.4.18 Stopping the FlashCopy Consistency Group 270
6.4.19 Stopping the FlashCopy mapping 272
6.4.20 Migrating between a fully allocated volume and a thin-provisioned volume . . . 272
6.4.21 Reversing and splitting a FlashCopy mapping 273

Chapter 7. Volume Mirroring 275
7.1 Volume Mirroring overview 276
7.1.1 Mirrored volume components 278
7.1.2 Volume Mirroring usage cases and characteristics 279
7.1.3 Performance considerations of Volume Mirroring 281
7.1.4 Bitmap space for out-of-sync volume copies 282
7.1.5 Synchronization status of volume copies 283
7.2 Volume Mirroring infrastructure design examples 283
7.2.1 HA design using a single SVC or Storwize V7000 cluster 283
7.2.2 HA and disaster-tolerant design using SVC or Storwize V7000 cluster 284
7.2.3 Quorum disks 286
7.2.4 Quorum disk considerations 286
7.2.5 High availability design using an SVC Stretched Cluster configuration 287
7.2.6 Non-ISL configuration 289
7.2.7 ISL configuration 294

Chapter 8. Implementing Volume Mirroring 299
8.1 Managing bitmap space for Volume Mirroring using the GUI 300

8.2 Managing bitmap space for Volume Mirroring using the CLI 302
8.3 Configuring Volume Mirroring using the GUI 304
8.3.1 Creating a new volume 304
8.3.2 Adding a mirrored copy to an existing volume 313
8.3.3 Deleting a mirrored copy from a volume mirror 316
8.3.4 Splitting a volume copy 317
8.3.5 Validating volume copies 318
8.3.6 Migrating to a thin-provisioned volume using Volume Mirroring 319
8.4 Configuring Volume Mirroring using the CLI 322
8.4.1 Creating a new volume 323
8.4.2 Adding a mirrored copy to an existing volume 324
8.4.3 Deleting a mirrored volume 328
8.4.4 Splitting a mirrored volume 329
8.4.5 Validating a mirrored volume 330
8.4.6 Configuring I/O Time-out for Mirrored Volumes 331

Chapter 9. Server integration 335
9.1 Copied data 336
9.2 Data on copied volumes 336
9.3 AIX specifics 337
9.3.1 JFS, JFS2, and Copy Services 337
9.3.2 Example of copy creation 338
9.3.3 Accessing useful copies in AIX 341
9.4 Windows specifics 350
9.4.1 Windows 2000 Server, Windows Server 2003, SVC Advanced Copy Services . 350
9.4.2 Microsoft Windows Server 2008 and IBM SVC Advanced Copy Services 367
9.5 Linux specifics 381
9.5.1 Creating useful copies 381
9.5.2 Accessing useful copies 384
9.5.3 Accessing target or secondary volumes from the same Linux host 384
9.5.4 Accessing target or secondary volumes from a different Linux host. 385
9.6 Other operating systems 386

Chapter 10. Automating copy services processes 387
10.1 Running commands on the cluster 388
10.1.1 Connection 388
10.1.2 Authentication 392
10.1.3 Submission 393
10.1.4 Authorization 393
10.1.5 Execution 393
10.2 Creating connections 393
10.2.1 Transient connections 394
10.2.2 Persistent connections 401
10.3 SVC command-line scripting 402
10.3.1 Submitting command-line scripts 403
10.3.2 Example SVC command-line script. 403
10.4 Server-side scripting 404
10.4.1 Creating persistent SVC connections 405
10.4.2 Handling the SVC response 411
10.5 SVC CIMOM programming 412
10.5.1 CIM concepts 414
10.5.2 SVC concepts mapping to CIM concepts 414
10.5.3 Creating and starting a FlashCopy mapping. 415

10.6 Logging 422
10.7 Auditing 424

Chapter 11. Software solutions and services based on IBM SVC Replication Family Services 427
11.1 IBM Tivoli Storage Productivity Center for Replication with IBM SVC Replication Family Services 428
11.1.1 TPC-R compared to SVC/SSPC console 428
11.1.2 Setting up TPC-R for use with the SVC 430
11.1.3 Configuring Copy Services relationships 433
11.1.4 Failover/Failback Metro and Global Mirror sessions 442
11.1.5 Metro and Global Mirror with practice sessions 443
11.1.6 Using CSV files during Copy Set creation. 443
11.1.7 TPC for Replication CLI 444
11.1.8 TPC-R Live demo 445
11.1.9 IBM Tivoli Storage Productivity Center V5.1 445
11.2 IBM Tivoli Storage FlashCopy Manager 446
11.2.1 IBM Tivoli Storage FlashCopy Manager for Windows. 447
11.2.2 Microsoft Exchange Server backup and restore with IBM Tivoli Storage FlashCopy Manager for Windows. 447
11.2.3 Microsoft File System backup and restore with IBM Tivoli Storage FlashCopy Manager for Windows. 458
11.2.4 IBM Tivoli Storage FlashCopy Manager for Oracle. 462
11.2.5 IBM Tivoli Storage FlashCopy Manager for VMware 467
11.2.6 IBM Tivoli Storage FlashCopy Manager live demonstration 468
11.3 IBM SVC plug-ins 468
11.3.1 IBM Storage Management Console for VMware vCenter. 468
11.3.2 Storage Management Pack for Microsoft System Center Operations Manager 470
11.3.3 IBM Storage Device Driver for VMware 470
11.4 IBM SVC/Storwize V7000 disaster recovery solution for VMware environment 470
11.5 IBM SVC/Storwize V7000 FlashCopy and Symantec NetBackup. 471
11.6 SVC/Storwize V7000 Remote Copy and Veritas Storage Foundation Enterprise HA 472

Chapter 12. Fabric design considerations for Replication Family 473
12.1 Topology and zoning for Metro and Global Mirror. 474
12.1.1 Dual WAN zoning 475
12.1.2 Single WAN zoning 475
12.1.3 Mirror-only ports 476
12.1.4 Alternate Global Mirror topology - “forwarding” I/O group. 476
12.2 FCoE and remote replication 478
12.3 Topology and zoning for Stretched Cluster. 479
12.3.1 Stretched Cluster and the third-site quorum requirement 479
12.3.2 “Classic” Stretched Cluster topology - no inter-node ISLs 479
12.3.3 Stretched Cluster with inter-node ISLs 481
12.4 WAN connectivity for replication 482
12.4.1 WAN connectivity types 482
12.4.2 WAN redundancy 483
12.4.3 Link quality 483
12.4.4 Bandwidth planning. 485
12.5 SAN routing with mirroring 487
12.5.1 SAN routing with Brocade products 488
12.5.2 SAN routing with Cisco Inter-VSAN Routes 490

Chapter 13. Troubleshooting Replication Family Services 491

13.1 Basic replication diagnostics 492
13.1.1 Common replication error codes 492
13.1.2 Collecting SVC performance data 493
13.2 Basic replication performance troubleshooting tips 494
13.2.1 First steps 494
13.2.2 Workload analysis 494
13.2.3 Source and destination storage controllers 494
13.2.4 SVC/Storwize V7000 CPU utilization 494
13.2.5 WAN performance (Metro and Non-Cycling Mode Global Mirror) 495
13.3 SAN diagnostics for replication 495
13.3.1 Fibre Channel flow control overview 495
13.4 FCIP congestion troubleshooting 500
13.4.1 Background 500
13.4.2 Checking WAN bandwidth configuration 501
13.4.3 FCIP tunnel quality check 505

Related publications 511
IBM Redbooks 511
Other publications 511
Referenced websites 512
Help from IBM 514

Notices

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

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

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

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

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

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

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

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

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

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

COPYRIGHT LICENSE:

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

Trademarks

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

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

AIX®
DB2®
DS4000®
DS6000™
DS8000®
Easy Tier®
Enterprise Storage Server®
FlashCopy®
Global Technology Services®
IBM®
pSeries®
Real-time Compression™
Redbooks®
Redbooks (logo) ®
Storwize®
System Storage®
Tivoli®
XIV®
xSeries®

The following terms are trademarks of other companies:

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

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

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

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

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

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

Summary of changes

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

Summary of Changes
for SG24-7574-02
for IBM® System Storage® SAN Volume Controller and IBM Storwize® V7000 Replication Family Services
as created or updated on February 16, 2017.

March 2013, Third Edition

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

New information

- New screenshots
- New algorithms
- New commands

Changed information

- Changed title
- Stretched Cluster Volume Mirroring was formerly known as Split Cluster Volume Mirroring
- Changed scripts
- Command usage changed

Preface

This IBM® Redbooks® publication describes the new features that have been added with the release of the IBM System Storage® SAN Volume Controller (SVC) and IBM System Storage Storwize® V7000 6.4.0 code, including Replication Family Services.

Replication Family Services refers to the various copy services available on the SVC and Storwize V7000 including IBM FlashCopy, Metro Mirror and Global Mirror, Global Mirror with Change Volumes, Volume Mirroring, and Stretched Cluster Volume Mirroring. The details behind the theory and practice of these services are examined, and SAN design suggestions and troubleshooting tips are provided. Planning requirements, automating copy services processed, and fabric design are explained. Multiple examples including implementation and server integration are included, along with a discussion of software solutions and services that are based on Replication Family Services.

This book is intended for use by pre-sales and post-sales support, and storage administrators. Readers are expected to have an advanced knowledge of the SVC, Storwize V7000, and the SAN environment.

The following publications are useful resources that provide background information:

- *Implementing the IBM System Storage SAN Volume Controller V6.3*, SG24-7933
- *Implementing the IBM Storwize V7000 V6.3*, SG24-7938
- *IBM SAN Volume Controller and Brocade Disaster Recovery Solutions for VMware*, REDP-4626
- *IBM System Storage SAN Volume Controller Upgrade Path from Version 4.3.1 to 6.1*, REDP-4716
- *Real-time Compression in SAN Volume Controller and Storwize V7000*, REDP-4859
- *SAN Volume Controller: Best Practices and Performance Guidelines*, SG24-7521
- *Implementing the Storwize V7000 and the IBM System Storage SAN32B-E4 Encryption Switch*, SG24-7977

The team who wrote this book

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

Jon Tate is a Project Manager for IBM System Storage SAN Solutions at the International Technical Support Organization, San Jose Center. Before joining the ITSO in 1999, he worked in the IBM Technical Support Center, providing Level 2 support for IBM storage products. Jon has 27 years of experience in storage software and management, services, and support, and is both an IBM Certified IT Specialist and an IBM SAN Certified Specialist. He is also the UK Chairman of the Storage Networking Industry Association.

Rafael Vilela Dias is a Senior IT Specialist working in IBM Global Technology Services® in IBM Brazil. He has 12 years of experience in the IT industry, mainly focused on IBM Storage System Solutions. He is certified for IBM Storage Subsystems and his areas of expertise include IBM, EMC, Hitashi, Fujitsu, Midrange and Enterprise Storage Systems, Brocade and Cisco SAN Network, IBM Tivoli® Storage Manager, xSeries® and pSeries® Servers. Rafael holds a degree in Computer Information Systems from Brazil, with academic works in Data Storage.

Ivaylo Dikanarov is a UNIX and Storage Technical Specialist for Integrated Technology Services in IBM UK. As part of the Infrastructure Management Centre he covers UNIX, Linux, Storage, SAN, Backup, Cluster, Grid and Cloud areas in planning, implementation and support activities for various IBM customers. Prior to joining IBM in 2008, Ivaylo spent more than 10 years working at middle and senior IT levels on enterprise infrastructure in the banking sector. His main professional interests are IT System Design, Implementation and Administration, IT Infrastructure Consolidation, Servers and Storage Virtualization, Networks, Security, Data Centers and Cloud services.

Jim Kelly is an IBM Certified Consulting IT Specialist working for the Systems and Technology Group in IBM New Zealand. He also blogs on IBM storage and is a member of the IBM social media team that engages with customers on SAN Volume Controller Ratings and Reviews at `http://www.ibm.com/systems/info/review.html`. Prior to joining IBM in 1999, Jim spent 13 years with Data General working with UNIX, storage, and network systems. His early career was spent on Data General minicomputers and as an operator in an IBM VSE mainframe environment.

Peter Mescher is a Product Engineer for the Solutions Central team, IBM Systems and Technology Group, Research Triangle Park, North Carolina, IBM US. He has many years of experience in IBM Storage and SAN troubleshooting, architecture, and implementation.

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

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

Pall Thorir Beck
Tom Jahn
Peter Mescher
Paulo Neto
Dan Rumney

There are many people who contributed to this book. In particular, we thank the development and PFE teams in IBM Hursley, UK.

We also thank the following people for their contributions:
Eric Bartlett
Chris Beeken
Chris Canto
Dave Carr
Gordon Cockburn
Robin Findlay
Carlos Fuente
Andrew Martin
Craig McAllister
Paul Merrison
Lucy Harris
Steve Randle
Bill Scales
Matt Smith
John Wilkinson
Barry Whyte
IBM Hursley, UK

Alexandre Chabrol
IBM France

Falk Schneider
IBM Germany

Mary Connell
Chris Saul
Bill Wiegand
IBM US

Special thanks to the Brocade Communications Systems staff in San Jose, California for their unparalleled support of this residency in terms of equipment and support in many areas:

Jim Baldyga
Silviano Gaona
Brian Steffler
Marcus Thordal
Steven Tong
Brocade Communications Systems

Now you can become a published author, too!

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

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

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

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

  ibm.com/redbooks

- Send your comments in an email to:

  redbooks@us.ibm.com

- Mail your comments to:

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

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks

- Follow us on Twitter:

  http://twitter.com/ibmredbooks

- Look for us on LinkedIn:

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

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

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

- Stay current on recent Redbooks publications with RSS Feeds:

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

1

Introduction to Replication Family Services

The term Replication Family Services refers to the various copy services available on the IBM System Storage SAN Volume Controller and IBM System Storage Storwize V7000.

The following topics are introduced in this chapter:

- IBM FlashCopy®
- Metro Mirror and Global Mirror
- Global Mirror with Change Volumes
- Volume Mirroring function
- Stretched Cluster (formerly Split Cluster) Volume Mirroring

Later chapters examine the details behind the theory and practice of these services, and also provide SAN design suggestions and troubleshooting tips.

1.1 FlashCopy

FlashCopy is a function that allows you to create a point-in-time copy of one of your SVC disks. This might be helpful when performing backups or application testing. These copies may be cascaded upon one another, read from, written to, and even reversed.

These copies are able to conserve storage, if needed, by being space-efficient copies that only record items that have changed from the originals instead of full copies.

1.2 Metro Mirror and Global Mirror

Metro Mirror and Global Mirror are technologies that enable you to keep a real-time copy of a disk at a remote site that contains another SVC Cluster or Storwize V7000 system.

Metro Mirror makes *synchronous* copies, which means that the original writes are not considered complete until the write to the destination disk has been confirmed. The distance between your two sites is usually determined by how much latency your applications can handle.

Global Mirror makes *asynchronous* copies of your disk. This means that the write is considered complete after it is complete at the local disk; it does not wait for the write to be confirmed at the remote cluster as Metro Mirror does. This greatly reduces the latency experienced by your applications if the other cluster is far away. However, it also means that during a failure, the data on the remote copy might not have the most recent changes committed to the local disk.

1.2.1 Global Mirror with Change Volumes

This function (also known as Cycle-Mode Global Mirror), introduced in SVC/V7000 V6.3, can best be described as “Continuous Remote FlashCopy.” If you use this feature, the SVC will essentially take periodic FlashCopies of a disk and write them to your remote destination.

This feature completely isolates the local copy from wide area network (WAN) issues and from sudden spikes in workload that might occur. The drawback is that your remote copy might lag behind the original by a significant amount, depending on how you have set up the cycle time.

1.3 Volume Mirroring function

Volume Mirroring is a function designed to increase high availability of the storage infrastructure. It provides the ability to create up to two local copies of a volume. Volume Mirroring may use space from two Storage Pools, and preferably from two separate back-end disk subsystems.

Primarily, you use this function to insulate hosts from the failure of a Storage Pool and also from the failure of a back-end disk subsystem. In the event of a Storage Pool failure, the cluster continues to provide service for the volume from the other copy on the other Storage Pool, with no disruption to the host.

You may also use Volume Mirroring to migrate from a thin-provisioned volume to a non-thin-provisioned volume, and to migrate data between Storage Pools of different extent sizes.

1.3.1 Stretched Cluster Volume Mirroring

The Volume Mirroring function also enables you to have real-time copies of a volume spread across two sites, but yet have those copies reside within the same SVC Cluster. This is useful for nearly instant disaster recovery and also for workload mobility solutions.

2

Planning for Replication Family Services

There are items that you need to consider during the planning phase for Replication Family Services. To be sure that we are delivering the best possible business value with SVC and Storwize V7000 technology we need to take a systematic approach starting from the business goals and considering the requirements and options, rather than moving directly into implementation activities.

> “Resist the urge to code” - Harlan D. Mills (1919-1996), IBM Fellow

The business goals that drive copy services are usually related to business continuity or productivity.

2.1 High-level design

It is useful to consider planning as an interactive process so that the initial statement of goals and requirements can be modified to take account of factors such as costs, availability of personnel and skills, technology features and limitations, and opportunities that arise.

2.1.1 Defining your project

The business goals that drive copy services might, for example, include supporting a business continuity strategy by improving storage high availability; the ability to restore lost or corrupted files; or the ability to recover volumes after a localized disaster. Goals might also include improvements to business productivity through more efficient storage practices in support of software development, business intelligence, and in general IT administration.

Understanding your current technical environment is an important step. For example, often your Fibre Channel fabric will need to be augmented as part of a replication services project, so you need to be clear about what is already in place. Also, several features related to SVC or Storwize V7000 have a host driver component, for example, both Tivoli Storage FlashCopy Manager and the new **`movevdisk`** command, so it is important to clearly document your host environment.

Also be clear about the success criteria that you are working towards. This will help everyone involved understand what needs to be done, why it is being done, and ultimately, whether the project is successful.

2.1.2 Functional requirements

Functional requirements are what you need the system to do in support of the business goals. Start from the requirements and then consider features that might help you to meet those requirements, rather than starting from the product features themselves.

An example of a functional requirement that might affect your implementation choices might be:

> "The transaction latency overhead, due to replication, on production writes should never exceed 5 ms."

Table 2-1 maps a range of functional requirements to IBM Replication Family features.

Table 2-1 Map of functional requirements to several IBM Replication Family features

Functional requirements	IBM Replication Family features
Create copies of volumes	FlashCopy, Volume Mirroring
Move volumes	movevdisk, Volume Mirroring, FlashCopy
Create copies of a consistency group of volumes	FlashCopy
Synchronously copy writes to a remote volume	Volume Mirroring, Metro Mirror
Synchronously copy writes to a remote volume or consistency group	Metro Mirror
Asynchronously copy writes to a remote volume or consistency group	Global Mirror

Functional requirements	IBM Replication Family features
Create hourly point-in-time copies of a volume or consistency group at a remote site	Global Mirror with Change Volumes
Restore remote volumes or consistency groups to the primary site on demand	Metro Mirror, Global Mirror, Global Mirror with Change Volumes
Restore a point-in-time copy of a volume or consistency group back onto the primary volume	Reverse FlashCopy
Ride-through the failure of a pool or disk system.	Volume Mirroring
Ride-through the failure of a pool, disk system, or site.	SVC Stretched Cluster
Provide application-consistent copies of data	FlashCopy in conjunction with Tivoli Storage FlashCopy Manager

2.1.3 Non-functional requirements

Non-functional requirements are the qualities that are important if the system is to support the business goals, rather than the specific functions that the system will perform.

The following general areas are considered to fall under non-functional requirements:

- Performance
- Capacity
- Availability
- Usability
- Security
- Privacy
- Maintainability
- Manageability
- Flexibility

An example of a non-functional requirement that might affect your implementation choices might be:

> “The transaction latency overhead, due to replication, on production writes should never impact application performance.”

Depending on the environment, a requirement like this could drive a choice for Global Mirror with Change Volumes (rather than traditional Global Mirror or Metro Mirror). Note that this is just an example, not a recommendation. Each customer environment will have its own set of non-functional requirements.

2.1.4 Prior to implementation

Having clarified the business issues to address, and having established a preliminary list of requirements, document any risks, assumptions, and dependencies. Consider the planned time frame, the costs and the availability of appropriate skills. Although information might be incomplete, industry experience might be enough to highlight potential issues such as the likely higher network costs for synchronous replication versus the lower network costs for asynchronous replication.

When you make decisions, for example, deciding between Global Mirror and Global Mirror with Change Volumes, it is best to document the options, the pros and cons of each, and the reasons for the choice you make. Documenting architectural decisions will encourage you to consider the options carefully and will also help others to understand why choices were made.

2.2 Business continuity and productivity goals

The business goals that drive copy services are usually related to business continuity or productivity. This translates to a range of options for data protection and features that allow you to manipulate storage volumes more efficiently.

2.2.1 Data protection

Business continuance maps to the copy services data protection capabilities:

- High Availability
- Backup and Restore
- Disaster Recovery

During planning, you need to ask a series of questions separately for each application data set. For example, you might decide that you need to protect your core transactional applications but not your data warehouse data sets, because they are easily regenerated.

Whenever data protection is discussed, you need to think in terms of Recovery Point Objective (RPO) and Recovery Time Objective (RTO).

Recovery Point Objective

Recovery Point Objective is defined as the amount of data loss, expressed in time units. Thus, an RPO of 1 minute means that when you recover from an outage, you have lost the data that was updated in the last minute before the problem struck.

Question to ask for each application: How much data loss can you bear?

Recovery Time Objective

Recovery Time Objective is defined as the amount of time it takes you to get back up and running. Understand, though, that having the storage system ready to go again is not the whole picture. It might take considerable additional time after that to recover network and server access and restore application integrity such as running fsck and replaying log files.

Question to ask for each application: How long can you afford to be offline?

These copy services capabilities are discussed in more detail in 2.3, “High availability” on page 9, in 2.4, “Backup and restore” on page 10 and in 2.5, “Disaster recovery” on page 12.

2.2.2 Storage efficiency

Business productivity maps to the copy services storage efficiency capabilities:

- Easy creation of volume copies
- Easy movement of volumes to a different pool
- Easy movement of volumes to a different I/O group
- Easy movement of volumes to a different cluster

These copy services capabilities are discussed in more detail in 2.6, “Creating volume copies” on page 17 and 2.7, “Moving volumes” on page 19.

2.3 High availability

A high availability (HA) solution is designed to avoid loss of access to data on your site, that is, RPO=0 and RTO=0. If the high availability mechanism acts quickly enough, then there is no application outage. The IBM copy services technology that maps most closely to this requirement is Volume Mirroring (formerly called vdisk mirroring) and also SVC Stretched Cluster.

> **Question to ask for each application:** Aside from “normal” high availability, what additional risks do you need to protect against, if any?

2.3.1 Volume Mirroring for high availability

With Volume Mirroring, the system can take up to 30 seconds to complete a volume failover when one or other copy of the data goes offline for any reason. This is fast enough to be within the limits of most operating systems tolerances.

Modern disk systems generally offer dual controllers, n+1 power and cooling, and RAID protection.

With Volume Mirroring you can further protect against:

- Complete failure of a storage pool (managed disk group)
- Complete failure of a virtualized disk system (for example, a DS3500, EVA P6300, or VNX5500)
- Any need to take a virtualized disk system offline for drive firmware upgrades

2.3.2 SVC Stretched Cluster for high availability

With Stretched Cluster you can further protect against the following scenarios:

- Failure of a computer room
- Failure of a building
- Failure of a campus
- A disaster centered up to 300 km away across a WDM network

One of the cost benefits of SVC Stretched Cluster is that there is no mirroring licence to buy. However, because Volume Mirroring is synchronous, fiber-optic latency is likely to restrict performance critical environments to distances below 100 km.

Additional functionality can be derived from combining SVC Stretched Cluster with VMware Vmotion or IBM AIX® Live Partition Mobility. VMware VMotion integration is discussed in “Inter-data center workload mobility and failover with VMware vSphere and IBM System Storage SAN Volume Controller,” which is available at:

http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101923

> **Use 2145-CF8 or CG8 nodes:** When planning a Stretched Cluster, use 2145-CF8 or CG8 nodes because these have 41 buffer credits per port. Earlier nodes support 8 buffer credits per port which mean, for example, a maximum distance of 4 km at 4 Gbps.

2.4 Backup and restore

Backup and restore is used to provide recovery from minor losses such as files that have been deleted, lost, or corrupted. A 24-hour cycle is usually considered acceptable, but granularity of restore and preservation of application integrity are important.

Backup and restore is also the final backstop in any broader disaster recovery plan. Restoring a whole site from backups would, however, likely be a long, slow process and one that might be prone to media errors. It also assumes there is a primary site in a working state to restore to.

Backup and restore most commonly refers to single file recovery with RPO=24 hours (although it can be less) and RTO=1 hour, perhaps depending on how long it takes your IT support staff to respond to a request for file recovery. Some systems allow user restoration of files.

2.4.1 FlashCopy for backup and restore

FlashCopy can be used to create backups to disk of volumes or consistency groups. To preserve the application integrity, you might need to purchase and deploy Tivoli Storage FlashCopy Manager in conjunction with FlashCopy. FlashCopy comes included with Storwize V7000, but is licensed per TB of source volume on SVC (not the total of source and target). By using FlashCopy you can reduce the RPO to a number of hours or minutes, depending on how often you are prepared to take point-in-time copies of your data.

Question to ask for each application: How much data loss can you bear? How often should you take FlashCopies?

FlashCopy volumes are write-order consistent (crash-consistent). Restarting from crash-consistent volumes is like restarting after a sudden power failure. There might be dependent writes in flight at any one time on a system, so a point-in-time copy might capture a system image half-way through a chain of dependent writes. This means that crash recovery routines will need to be run, for example, fsck, replaying database log files, before the volumes can be used normally.

Tivoli Storage FlashCopy Manager can greatly simplify backup and recovery using FlashCopy. Tivoli Storage FlashCopy Manager gets around the application consistency issues by integrating with applications such as Oracle, IBM DB2®, SAP, VMware and other custom applications to flush and pause I/O before triggering the FlashCopy. If you use FlashCopy with Tivoli Storage FlashCopy Manager and Tivoli Storage Manager you can provide an integrated application-to-disk-to-tape backup and restore solution.

Figure 2-1 on page 11 shows Tivoli Storage FlashCopy Manager being used on a data store through VMware vCenter.

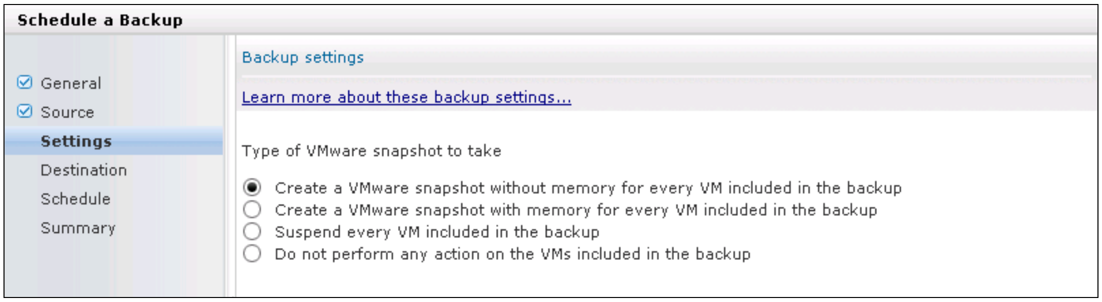


Figure 2-1 Tivoli Storage FlashCopy Manager operation on a VMware datastore

FlashCopy supports making a cascade of up to four incremental copies of a volume, for example, for backups. If you need more than four FlashCopy mappings, then the FlashCopy mappings will lose their incremental nature and become full FlashCopies.

Metro Mirror and Global Mirror secondary volumes can also be used as crash-consistent backup volumes. However, they might also require recovery routines before volumes can be used for production. At the time of writing, Tivoli Storage FlashCopy Manager did not integrate with Metro Mirror and Global Mirror.

Question to ask for each application: Do you shut down your applications before taking FlashCopies, or use Tivoli Storage FlashCopy Manager to ensure application consistency?

2.4.2 Using Reverse FlashCopy to restore a volume

You can create up to 256 FlashCopies of a volume, and you can use a FlashCopy to restore a volume without destroying the original FlashCopy mappings. There are several unique capabilities that make this particularly useful.

- You can restore a source volume from a FlashCopy even while the FlashCopy is mounted somewhere else, for example, to a backup server that is copying it to tape.
- You can create a FlashCopy of a corrupted production source volume before you restore from the backup FlashCopy, so that you retain the corrupted data for analysis
- You can set up the reverse FlashCopy mapping in advance so that it can be easily triggered as needed.

Example 2-1 shows how to set up the reverse FlashCopy mapping in advance.

Example 2-1 Establishing a reverse FlashCopy mapping

```
mkfcmap -source vol1 -target vol0 —name reversefcmap1
```

Example 2-2 shows how to trigger the reverse FlashCopy.

Example 2-2 Triggering a reverse FlashCopy

```
startfcmap -prep -restore reversefcmap1
```

Note the following points:

- **`-prep`** flushes any vol1 cache entries to disk, sets vol1 to write-through mode, and discards any vol0 cache entries.

- The reverse FlashCopy only copies back the grains that are needed to restore vol0 to the given point in time.
- After the reversefcmap1 status is `copying`, the application I/O can resume to vol0 while data is copied from vol1.
- If you have multiple options of FlashCopy volumes to restore from, you can also stop a restore in progress and start another one.

Example 2-3 shows how to stop a reverse FlashCopy restore operation that is in progress.

Example 2-3 Stopping a reverse FlashCopy restore process

```
stopfcmap -force reversefcmap1
```

2.4.3 Backup and restore references

For more information about backup and restore, consult the following documentation:

- Storwize V7000 with Tivoli FlashCopy Manager: Backup Solution for Exchange 2010

  http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101764
- Protecting VMware data with Tivoli Storage FlashCopy Manager for VMware and Tivoli Storage Manager for Virtual Environments

  http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP102021
- SAP with IBM Tivoli Storage FlashCopy Manager for VMware, IBM XIV® and Storwize V7000

  http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP102093
- Protecting Oracle RAC ASM databases with IBM Tivoli Storage FlashCopy Manager

  http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP102113
- Storwize V7000 Practice guide: Backup & restore of Oracle Database using Tivoli FlashCopy Manager

  http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101771
- SVC 6.3 Copy Services for Backup & Recovery of Oracle 11.2 RAC/ASM Databases

  http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP102080
- Using Symantec NetBackup and IBM Storwize V7000 FlashCopy/Concepts and best practices for backup and restore solution on Linux

  http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101767
- Epic and Storwize V7000 - Solution overview and performance benchmark with FlashCopy

  http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP102011

2.5 Disaster recovery

Maintaining a warm site with a copy of your data ready to go is the usual way of ensuring rapid recovery from a large-scale failure.

Figure 2-2 on page 13 provides a timeline of a typical IT disaster recovery (DR).

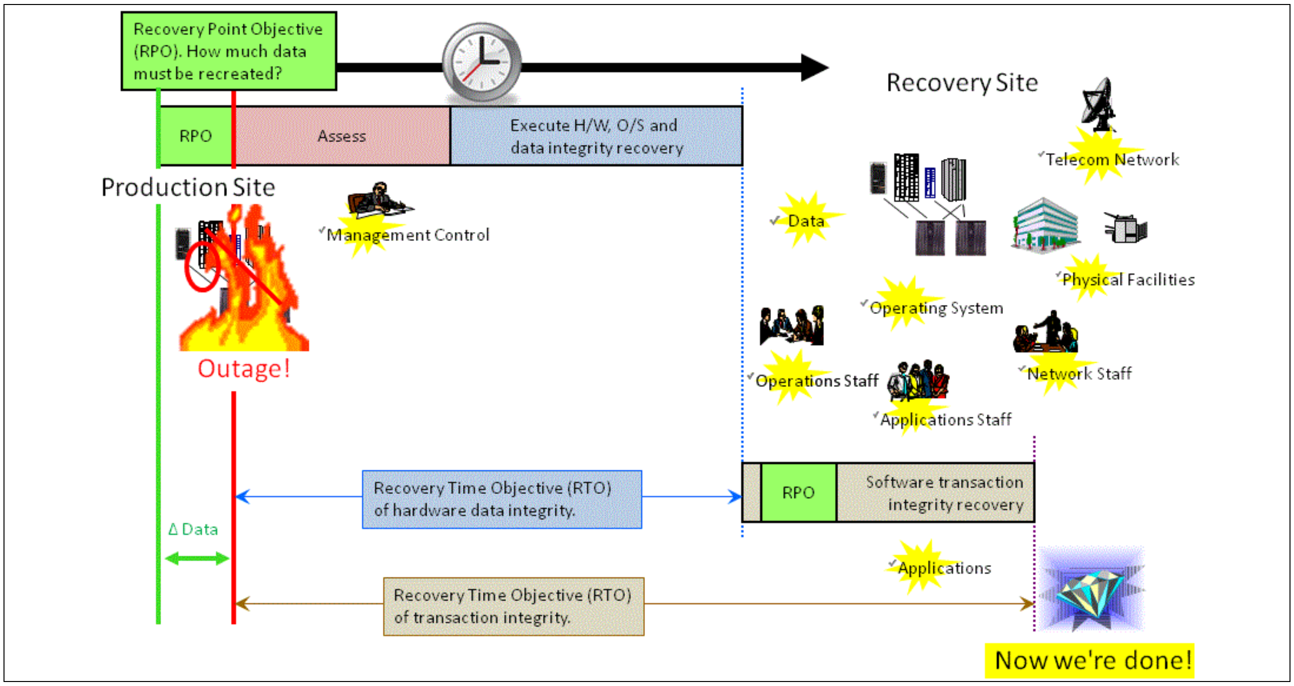


Figure 2-2 Timeline of an IT Recovery

The IBM copy services technologies that map most closely to a disaster recovery requirement are Metro Mirror, Global Mirror, and Global Mirror with Change Volumes. SVC Stretched Cluster is also an option in some situations.

Along with RPO and RTO, consider budgets when undertaking DR planning. The cost of network bandwidth, for example, can vary significantly by country or region, and this might affect your choice of SVC Stretched Cluster, Metro Mirror, Global Mirror, or Global Mirror with Change Volumes. Licensing differences between SVC Stretched Cluster and Metro and Global Mirror can also be a factor.

> **Question to ask for each application:** Which technology choice provides the most cost-effective way to meet the recovery requirements?

Keep in mind that, although the copies of data produced are designed to be crash-consistent, recovery routines such as fsck and log replays might need to be run on them before applications can be used.

2.5.1 Disaster recovery site location

When creating a DR solution, one of the key considerations is the geographical separation between the primary site and the DR site.

In 2003, the US Securities and Exchange Commission (SEC) suggested that 200 miles (320 kms) would be appropriate between the primary and secondary sites for a US financial institution.

Many commercial and governmental companies are, however, content with distances of 30 km or even less as a compromise between cost, convenience and protection, because most outages have relatively localized causes.

The following list includes examples of disaster radii you might want to consider:

- Regional power outage
- Problem localized to computer room or building or urban area
- Typical hurricane 150 km
- 2005 Hurricane Katrina, 250 km sweep
- 2011 Thailand floods, up to 60 km radius
- 2011Christchurch earthquake, 60 km damage radius
- 2011 Tohoku earthquake and tsunami, 10 km inland
- 2011 Fukushima Daiichi disaster, up to 50 km sweep
- 1980 Mt St. Helens volcanic eruption, up to 30 km radius

Questions to ask: What kind of disasters do I need to protect against?

2.5.2 Metro Mirror for disaster recovery

Metro Mirror is good fit for a zero RPO requirement if the network is not a significant limiting factor, and assuming the inherent extra transaction latency of a synchronous replication can be tolerated.

A key part of planning for Metro Mirror will be to minimize the application performance impact. You will need a low latency FC or FCoE network with enough bandwidth to keep up with peak production writes, and a disk system at the auxiliary site that can also keep up with peak writes. The network connection may be carried by wave division multiplexors. Planning to use Metro Mirror over an FCIP connection is not advisable.

Question to ask for each volume: Do I have enough network and auxiliary disk performance to support low latency copying at peak times?

2.5.3 Global Mirror for disaster recovery

Global Mirror is best used where the desired RPO is near-zero if the network is not a significant limiting factor, and assuming the minor application performance impact of tag and release processing on each write can be tolerated.

A key part of planning for Global Mirror will be to minimize the application performance impact, and thereby avoid GM choosing to stop the busiest relationships. You will need enough bandwidth to keep up with the production writes and a well-performing disk system at the auxiliary site. You will need an FC, FCoE, or FCIP network and a well-performing disk system at the auxiliary site. The network connection may be carried by wave division multiplexors.

Question to ask for each application: Do I have enough network and auxiliary disk performance to ensure the remote copy relationships can be sustained during peak times?

2.5.4 FlashCopy in a Metro or Global Mirror recovery

You can use FlashCopy to create a consistent copy of an image before you restart a Global Mirror relationship.

When a consistent remote copy relationship is stopped and if production work continues, the DR site will fall out of sync. When you restart the relationship, the resync process is not write-order consistent, so you will not have a consistent copy back at DR until the resync has completed. It is generally prudent, therefore, to take a FlashCopy of them in their consistent but out-of-date state prior to beginning the resync, so that if anything goes amiss during the time it takes to complete the resync you will at least still have a consistent copy of the data at DR to work from.

Similarly, if you want to perform testing of your DR copy without suspending the replication, you might need to take a FlashCopy of the volumes, which you can then mount and test on with DR servers.

After a disaster is over, this approach can also be used before copying your DR volumes back to production. If your production volumes were not actually damaged in the disaster it is prudent to take a FlashCopy of them in their consistent but out-of-date state prior to beginning the resync from DR.

Scripting tools for SAN Volume Controller available from the following site `https://www.ibm.com/developerworks/mydeveloperworks/groups/service/html/communityview?communityUuid=5cca19c3-f039-4e00-964a-c5934226abc1` include a `copymanager` script that shows how to manage the FlashCopies in this situation.

If you are using SAN Volume Controller, this would mean ensuring that you purchase FlashCopy licences (FlashCopy licences are not charged separately on Storwize V7000).

These steps are not required when using Global Mirror with Change Volumes because consistent FlashCopy volumes are automatically created.

2.5.5 Global Mirror with Change Volumes for disaster recovery

Global Mirror with Change Volumes might be the best choice when the RPO can be higher (the default is 10 minutes) and the best application performance is desired. This choice can reduce the network requirements, and host performance will be independent of the replication.

A key part of planning for Global Mirror with Change Volumes is to ensure that there is sufficient network bandwidth to meet the planned RPO. Note that RPO is calculated as not more than 2 x the cycle period, assuming there is enough network bandwidth to complete the copying inside the cycle period.

You will need an FC, FCoE, or FCIP network and enough disk space at the auxiliary site. The network connection may be carried by wave division multiplexors.

Auxiliary disk system performance might affect the RPO, but it cannot affect host application performance. Disk performance sizing will, however, need to take into account the expected workload to be placed upon it in a disaster scenario.

Questions to ask for each application:

- Do I have enough network bandwidth to meet my planned RPO?
- Am I prepared to miss some cycle periods at peak times and accept a correspondingly larger RPO?
- Will my auxiliary disk system provide enough performance if it is ever used to run production applications?

2.5.6 SVC Stretched Cluster for disaster recovery

SVC 6.3 introduced higher distance limits on Stretched Cluster configurations and the ability to place the primary quorum at a remote location over an FCIP link. With these enhancements, SVC Stretched Cluster is now a more complete disaster recovery option.

Table 2-2 is designed to help you choose between SVC Stretched Cluster, Metro Mirror, Global Mirror and Global Mirror with Change Volumes.

Table 2-2 Comparison of Metro and Global Mirror to SVC Stretched Cluster

	SVC Stretched Cluster	**Metro Mirror**	**Global Mirror**	**GM with Change Volumes**
Failover	Automatic	Manual	Manual	Manual
Interruption to Service	No	Yes	Yes	Yes
Resynchronization	Automatic	Manual	Manual	Manual
RPO	Zero	Zero	Near-zero	Minutes/hours
RTO	< 30 seconds	Minutes/hours	Minutes/hours	Minutes/hours
Main network options	FCP, FCoE	FCP, FCoE	FCP, FCoE, FCIP	FCP, FCoE, FCIP
Network bandwidth				
Distance limit	300 km	300 km	8000 km	8000 km
Additional licensing	None	per TB, both ends	per TB, both ends	per TB, both ends
Third site (Power domain) quorum	Required	No	No	No
Fabric configuration	Specific	Standard	Standard	Standard

Question to ask: Is SVC Stretched Cluster a more cost-effective way to achieve your disaster recovery goals?

Table 2-3 shows the supported FC distances for SVC Stretched Cluster.

Table 2-3 Supported FC distances for SVC Stretched Cluster

Maximum distance	**Usage**	**Requirements**
10, 20, 40 km	Failover, Vmotion, Live Partition Mobility	8 Gbps, 4 Gbps, 2 Gbps FC and traditional Stretched Cluster rules.
100 km	Failover, Vmotion, Live Partition Mobility	WDM. Private ISLs between nodes.
300 km	Failover	WDM. Private ISLs between nodes.

Tip: Using ISLs between nodes can improve performance significantly even for distances well below 40 km.

SVC Stretched Cluster is based on synchronous Volume Mirroring, so latency and buffer credits need to be considered.

Tip: The 8 Gbps FC adapters on the SVC 2145-CG8 nodes support 41 buffer credits per port. You might need to set your FC switch port buffer credits to 41 also. The SVC will not auto-negotiate this for you.

2.5.7 Three-site replication for disaster recovery

You can also combine SVC Stretched Cluster with Global Mirror to build a three-site replication solution. Three-site replication is a way to address a need for both a synchronous copy with automated failover and a long-distance separation requirement. This might be more cost-effective and perform better than trying to create a long-distance synchronous copy. Remember, however, that a three-site solution will need three times the production disk space.

Question to ask: Do I need the benefits of both a synchronous copy with automated failover and a long-distance copy of data?

2.5.8 Further reading for disaster recovery

- Interdata center workload mobility and failover with VMware vSphere and SAN Volume Controller Stretched Cluster

  http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101923
- Disaster recovery for SAP with VMware Site Recovery Manager on IBM Storwize V7000

  http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101913
- SVC Global Mirror - a practical review of important parameters

  http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101848

2.6 Creating volume copies

Creating full data copies might be useful in a data mining, development, testing, or training environment where a production volume is the source of the copying. The copy technologies that can be used to meet this requirement are Volume Mirroring (formerly known as VDisk mirroring), FlashCopy, and Metro or Global Mirror.

2.6.1 Single Volume Mirroring

Volume Mirroring creates a local synchronous mirror, typically on a separate pool. You need to be careful that the second pool can keep up with the write activity on the original pool, plus additional background synchronization, which defaults to 2 MBps.

For example, if you have a production volume in a storage pool built on 46 x 10 K RPM drives in RAID10 and you create a local mirror using **addvdiskcopy** and specify the copy to be in a pool built from 11 x 7200 RPM drives in RAID6, writes are mirrored to both volumes and the application might be limited by the speed of slower volume.

By default, if one of the copies takes longer than 5 seconds to complete a write, the slow write is aborted and that copy is allowed to go out of sync. The write to the other copy is marked as complete (a cleanup process that should complete in less than 25 seconds). The system will then stop using the slow copy for around 5 minutes before trying to resync. If you do not plan your volume performance properly, this can happen repeatedly.

When you create a volume copy you can optionally specify **`-mirrorwritepriority redundancy`**, which will force the two to stay in sync even when poor performance on one copy is making the application run slowly. This is not normally advised.

After the two copies have fully synchronized you can pause all write I/Os to the volume pair and issue **`splitvdiskcopy`** and end up with two separate volumes containing identical data. If you issue **`splitvdiskcopy -iogrp`**, you can also specify which I/O group in your clustered system you want to own the new volume.

If you do not pause the application that is writing to the volumes before you split the mirror, the split copy will be crash consistent (write order consistent) but not application consistent. Recovery routines (for example, fsck and replay of log files) might need to be run on the volume before it can be used.

> **Managing the synchronization rate:** Adding a mirror to a production volume will create additional read workload on the production volume during synchronization. The synchronization rate can be managed using **`addvdiskcopy -syncrate`**.

2.6.2 FlashCopy (volumes and Consistency Groups)

FlashCopy can also be used to create a copy of a volume. Whereas Volume Mirroring copies the data first, and then creates the point in time split, FlashCopy will start with a point in time split and then copy the data. FlashCopy also offers you the option of working with a group of volumes if you want to maintain write order consistency across the group, for example, a database volume and a logs volume.

By default, FlashCopy creates a full copy of a volume, typically on a separate pool and optionally on a separate I/O group within the same clustered system. The same general performance considerations apply to FlashCopy as are described in 2.6.1, “Single Volume Mirroring” on page 17. The background copy rate can be managed using **`chfcmap -copyrate`**.

After the two copies have fully synchronized, you can issue **`stopfcmap`** or **`stopfcconsistgrp`** and end up with two independent volumes. If you do not pause the application that is writing to the source volumes before you start the FlashCopy, the FlashCopy volumes will be crash consistent (write order consistent) but not application consistent. Recovery routines (for example, fsck and replay of log files) might need to be run on the volumes before they can be used.

FlashCopy also provides the option to create a copy-on-write volume without any additional background copying. In this case the FlashCopy volume will remain dependent on the source volume. Reads for most blocks will be passed straight back to the source volume. This can impact the production volume performance each time the FlashCopy volume is accessed. For example, running data mining processes off a copy-on-write dependent FlashCopy is likely to impact production performance. Similarly, backup of a dependent FlashCopy volume to tape is likely to place significant read loads on the production volume.

> **Read consideration:** Reads to the first dependent FlashCopy volume in a chain can place read loads onto the source volume.

2.6.3 Metro or Global Mirror

Metro or Global Mirror can also be used to create a copy of a volume either on the same I/O group (intracluster Metro Mirror) or to a partnered cluster. The planning involved is a subset of that described for Metro or Global Mirror in 2.5, “Disaster recovery” on page 12.

2.7 Moving volumes

Moving volumes can be useful for load balancing. The three techniques that can be used are migration, create-mirror/split-mirror, and the new **movevdisk** command. (As part of planning for a volume move, it is useful to try it on a test volume.)

Questions to ask for each volume move: Do I need to move the volume to a new storage pool, to a new I/O group, or to a new cluster?

2.7.1 migratevdisk

The **migratevdisk** command is used to move a volume from one storage pool to another, within the same I/O group. This is an online in-flight migration that has been available since the early introduction of SVC. Its use is comprehensively covered in *Implementing the IBM Storwize V7000 V6.3*, SG24-7938, and *Implementing the IBM System Storage SAN Volume Controller V6.3*, SG24-7933.

2.7.2 Volume Mirroring usage

Volume Mirroring can be used to move a volume in the sense of creating a mirror of a volume as described in 2.6.1, “Single Volume Mirroring” on page 17, and then simply deleting the original mirror or copy. There is no need to issue **splitvdisk** in this case. This achieves the same end as using **migratevdisk.**

If you want to move the volume to a different I/O group, rather than simply to a different storage pool, you can pause all write I/Os to the volume pair and issue **splitvdiskcopy -iogrp** to specify which I/O group in your clustered system you want to own the new volume.

If you do not pause the application that is writing to the volumes before you split the mirror, the split copy will be crash consistent (write order consistent) but not application consistent. Recovery routines, for example fsck and replay of logs, might therefore need to be run on the volume before it can be used. Having moved the volume to a new I/O group, you might need to check your zoning and multi-path driver setup to ensure that your host application can see the new volume.

Outage consideration: If you plan to move a volume to a new I/O group using Volume Mirroring, you might need to take a temporary outage on your application. Using **movevdisk** instead might allow you to do this nondisruptively.

2.7.3 movevdisk

The movedisk feature, new in Version 6.4, is dependent on predictable intelligent behavior from the multipath driver. It will initially have limited support until thorough testing can be completed (for example, unclustered, Red Hat 6.x, SUSE Linux Enterprise Server 11, and possibly Microsoft Windows 2008, attached through FC or FCoE). Check the latest documentation before using this new feature.

The **movevdisk** feature is not currently supported with iSCSI and is not supported with clustered hosts. Various clusters, for example, VMware, require that the SCSI LUN ID does not change on a volume. At the time of writing **movevdisk** cannot force the SCSI LUN ID to stay the same after a move. Check the latest documentation for possible updates to this functionality.

Figure 2-3 shows the new option to move a volume to another I/O Group.

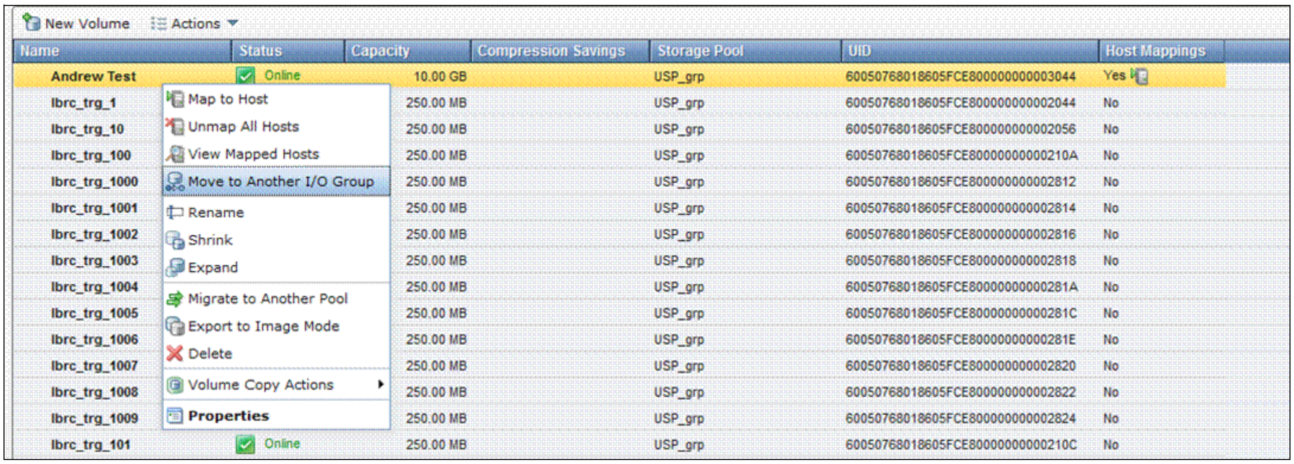


Figure 2-3 Moving a volume to a different I/O group

> **Possible temporary outage:** If you plan to move a volume to a new I/O group, you might need to take a temporary outage on your application, depending on your operating system environment.

2.7.4 Metro or Global Mirror

Metro or Global Mirror can also be used to move a volume either on the same I/O group (intracluster Metro Mirror) or to a partnered cluster. The planning involved is a subset of that described for Metro or Global Mirror in 2.5, “Disaster recovery” on page 12.

> **Temporary outage:** If you plan to move a volume using Metro or Global Mirror, you will need to take a temporary outage on your application.

2.8 Capacity planning

When planning copy services you need to understand your controller headroom, your write rates and sizes, and the space available for data copies.

When sizing a Storwize V7000 or SVC system, your IBM Business Partner should be able to access the IBM Disk Magic tool from IBM Partnerworld to model the performance of the system, assuming you have reasonably accurate information about your current and planned workload.

Information about your current workload can be drawn from a number of sources. In a fully virtualized server environment, you can, for example, draw information about write rates and average I/O sizes from VMware capacity planner reports, although you would need to enter summary information manually into Disk Magic.

Storage-based monitoring tools are generally better at giving you information across the whole server environment. However, if you are migrating from non-IBM storage, you might not be licensed for that vendor's monitoring tools. Although you can also run Windows perfmon or UNIX or Linux iostat reports and import that information directly into Disk Magic, be careful to follow the command options exactly if the output files are to be readable by Disk Magic. Also be aware that each host report will only give you a small subset of storage information in a large complex environment.

If you are already running Storwize V7000 or SVC, there are a number of tools available for performance monitoring.

2.8.1 Performance monitoring on SVC and Storwize V7000

There are a range of options for monitoring performance on SVC and Storwize V7000.

SVC and Storwize V7000 built-in monitor

There is a simple performance monitor built into SVC 6.2 and later that will provide various information in real time. This monitor is available from the GUI or through the CLI command **`lsnodestats`**.

Figure 2-4 shows the built-in monitor, which has been included since code level 6.2.

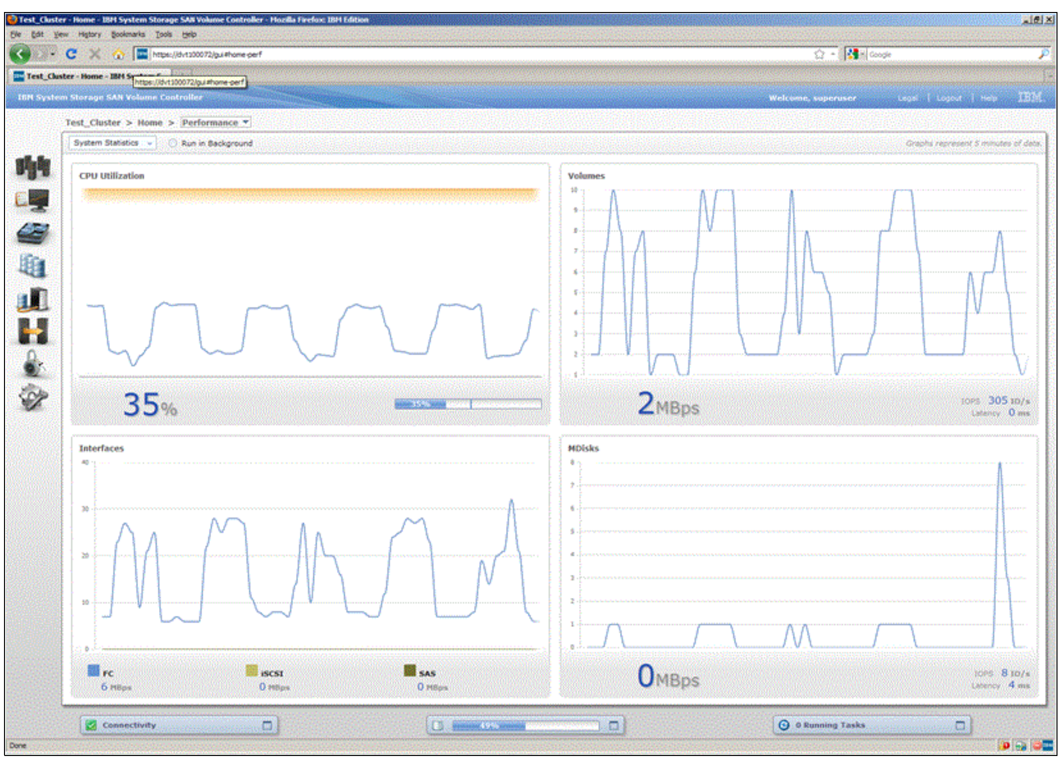


Figure 2-4 SVC and Storwize built-in performance monitor

Tivoli Storage Productivity Center 5.1

The most sophisticated of the performance monitoring options is Tivoli Storage Productivity Center (TPC), which will collect a wide range of metrics at minimum intervals of five minutes and store them in a data mart for detailed reporting.

Figure 2-5 shows the new feature structure for TPC 5.1

		TPC 4.2						
	Features	Basic Edition	Disk	Data	Repl	Standard Edition	TPC 5.1	VSC
Basic	Discovery, topology, monitoring, capacity management,alerting	●	●	●		●	●	●
Disk	Device Performance Management		●			●	●	●
Repl	2-site Replication Management				●		●	●
	3-site Replication Management				●		●	●
Data	Basic Capacity Reporting, NAS Discovery			●		●	●	●
	File System, DB Scans, Analytics, Data placement, quotas, TSM int			●		●		✔
Standard Edition	Fabric Performance Management					✔	●	●
	Storage Optimizer					✔		✔
	Configuration Change Management and best practices					✔		✔
	SAN Storage Planners & Policy based management					✔		✔
	Storage Tier Reports					✔		✔
New	Tiered Storage Optimization (Sage)							✔
	Cloud Service API						●	●
A-la-carte	FlashCopy Manager Software							●
	SAN Volume Controller Software							●

Figure 2-5 TPC 5.1 and TPC Virtual Storage Center

svcmon

svcmon is an unofficial (free and unsupported) monitoring tool for SVC and Storwize V7000 created by Yoshimichi Kosuge. It is designed to be run for a single period of 24 hours, collecting data every minute for analysis and exporting data in csv format, and optionally generating graphs in GIF format. Version 0.6c was released in May 2012 and includes usability enhancements. svcmon is a fairly technical product to install because it requires postgreSQL, perl, and puTTY.

svcmon is available for download from:

`https://www.ibm.com/developerworks/mydeveloperworks/blogs/svcmon`

qperf

qperf is an unofficial (free and unsupported) collection of `awk` scripts that was made available for download in May 2012 from IBM Techdocs. It was written by Christian Karpp and is designed to provide a quick performance overview using the CLI and a UNIX korn shell (it can also be used with Cygwin).

qperf is available for download from:

`http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/TD105947`

2.8.2 Using Disk Magic to predict the impact of Metro and Global Mirror

Disk Magic includes options to account for the performance overheads of Metro or Global Mirror.

Here is an example of a heavily loaded Storwize V7000 6.3 replication:

- 20 Km site separation with 2 x 8 Gbps links
- A single pool of disk with a nett capacity of 71.2 TB
  - 156 x 900 GB 10 KRPM 2.5" SAS drives in RAID10
  - 4x SSD 400 GB drives in RAID5, using IBM Easy Tier® with skew level 14
  - Containing 60 TB of data (that is, 84.3 percent full)
- 16 KB average I/O size
- 70/30 read/write ratio

We need to change replication settings in two places in Disk Magic. First, on the interfaces tab, and second, on the **Remote Copy** button on the **Open Workload** tab.

On the **Interfaces** tab, select **PPRC** for Metro Mirror (Peer to Peer Remote Copy is the older terminology) or **PPRC (XD/Async)** for Global Mirror.

On the **Open Workload** tab, select the **Remote Copy** button and set the Remote Copy Percentage which is the percentage of writes that will be remote copied. Our example uses 100%, but most real-life situations will have a lower percentage than that. You also need to again choose whether you want sync or async.

Figure 2-6 on page 24 shows an example line graph, automatically generated from Disk Magic, of the source volume latency impact from Metro or Global Mirror.

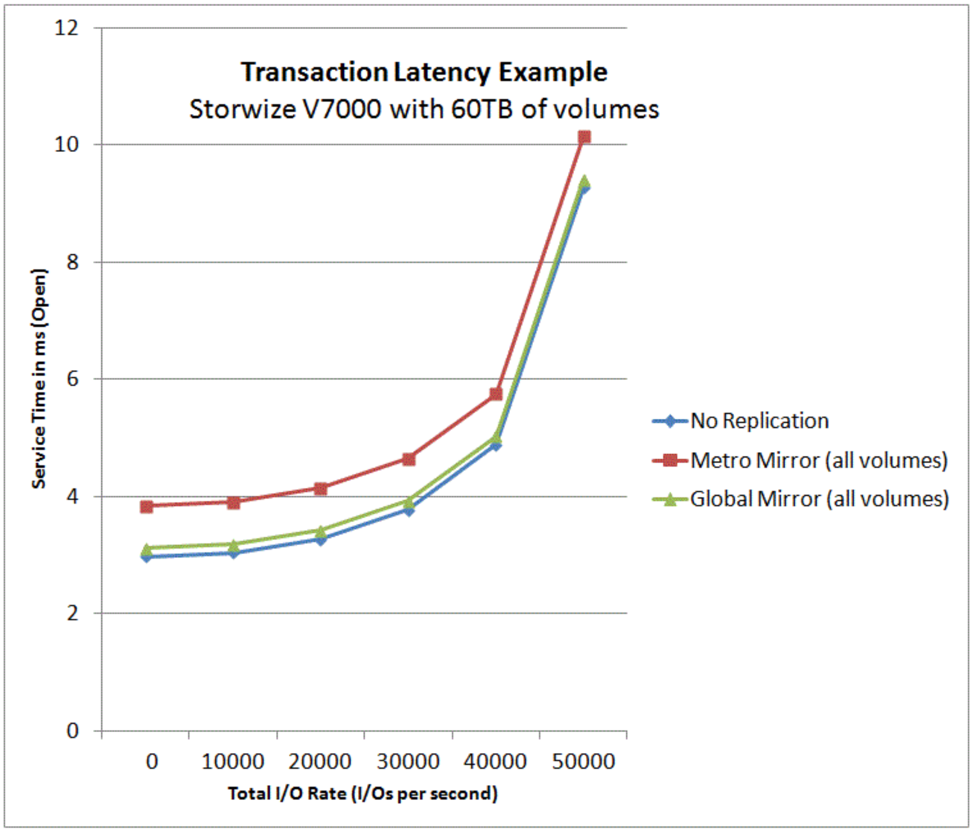


Figure 2-6 Transaction latency impact from Metro and Global Mirror

The transaction latency graph indicates you can expect a small overhead impact from Global Mirror, and slightly more from Metro Mirror.

Global Mirror with Change Volumes is not represented on the graph because the overhead from GM/CV should be effectively zero (0). There is no per transaction latency as such, only the general effect that running regular FlashCopies will have on the system's CPU and cache resources.

We also examine the Utilization tab in Disk Magic to observe the predicted changes in processor and bus utilization. Figure 2-7 on page 25 shows the utilization tab in Disk Magic.

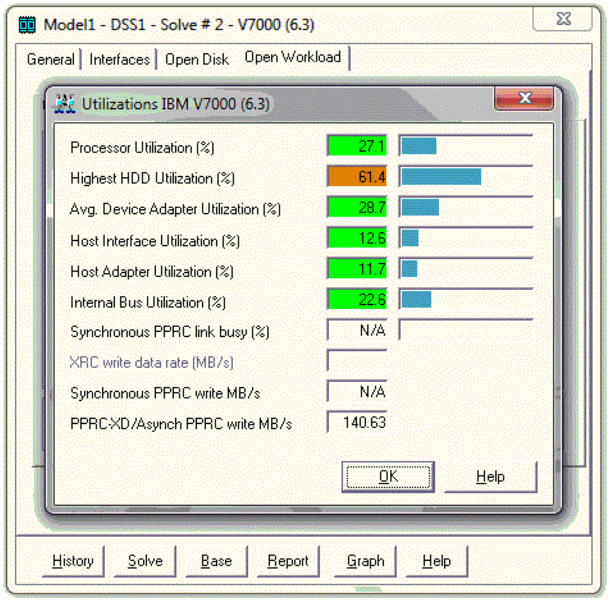


Figure 2-7 Disk Magic utilization tab

The write MBps figure also provides a prediction for your network sizing, in this case 140.63 MBps plus background copy allowance plus partnership heartbeat plus safety margin plus growth, for Metro or Global Mirror. However, bandwidth could be somewhat lower if Global Mirror with Change Volumes were used.

When it comes to allowing bandwidth for the partnership heartbeat, if you allow 0.8 Mbps for each node (that is, 1.6 Mbps for each I/O group) in a partnership, that will be generous. For example, for a basic 2-node to 2-node partnership allow 4 x 0.8 = 3.2 Mbps. For an 8-node to 8-node partnership allow 16 x 0.8 = 12.8 Mbps. The actual requirements are slightly lower than this.

Table 2-4 shows the example solution metrics. Note that the utilization figures for HDD and device adapters do not change between the options.

Table 2-4 Utilization metrics for our example solution

Utilization	Without replication	Global Mirror 20 Km 60 TB replicated	Metro Mirror 20 Km 60 TB replicated
Processor %	14.0	27.1	34.5
Host Interface	8.6	12.6	14.0
Host Adapter	6.9	11.7	20.0
Internal Bus	12.6	22.6	24.8
Replication MBps	0	140.6	140.6

2.8.3 FlashCopy considerations for disk performance

When planning for FlashCopy, be aware that a live production volume cannot proceed with the first write to any given block until it has copied the old block to its FlashCopy target if there is one. This is known as *Copy on Write*.

If you place your FlashCopies on slow disk, it could potentially impact the performance of a busy production application. There are, however, various ways to implement Copy on Write. In the case of SVC and Storwize V7000, when there are multiple FlashCopy maps from one source volume, the system will use the newest one as its primary map. It will delegate that target with the task of populating the next newest, and so on in a cascaded chain, even though the mappings seem to all be back to the source volume. Therefore, the source volume does not have to synchronously update all of the target volumes. This is designed to reduce the load on a source volume, which is why SVC can support up to 256 FlashCopy mappings per source volume.

The rate at which each FlashCopy target updates the next in the chain is called the *cleaning rate*. This rate can be set using the **`-cleanrate`** parameter on the **`mkfcmap`** command. The default value is 50, which is equivalent to 2 MBps. The scale is logarithmic, so reducing **`-cleanrate`** from 50 to 40, for example, will halve the throughput to 1 MBps.

2.8.4 Grain size

When data is copied between volumes, it is copied in chunks known as *grains*.

- The Metro Mirror and Global Mirror grain size is always 256 KB.
- The FlashCopy grain size is specified at the time the **`mkfcmap`** is issued. It can be 64 KB or 256 KB, with 256 KB being the default.

Grain size affects how much memory is used to manage the copy relationship, as described in the following table.

Table 2-5 Grain size affects the amount of system memory that is used

Function and grain size	Amount of RAM per 1 TB of source volume
Metro or Global Mirror 256 KB	0.5 MB
Volume Mirroring	0.5 MB
FlashCopy 256 KB (default)	0.5 MB
FlashCopy 64 KB	2.0 MB
Incremental FlashCopy 256 KB (default)	1.0 MB
Incremental FlashCopy 64 KB	4.0 MB

Because a Metro or Global Mirror volume can be reversed, bitmap space needs to be allocated on both systems in the partnership.

When using FlashCopy to copy a full volume, using a 256 KB grain size is more efficient. However, every write to the source volume will result in a copy-on-write to the FlashCopy target volume unless that grain has previously been copied to the target. This happens synchronously, so if you issue a 4 KB write to the source be mindful that this might result in a synchronous 64 KB or 256 KB write to the FlashCopy target.

Table 2-6 on page 27 summarizes the various implications of choosing either 64 KB or 256 KB grain size for the FlashCopy mapping.

Table 2-6 Implications of choosing 64 KB or 256 KB grain size for FlashCopy

	Memory usage	Synchronous write	Background copy	Target space usage
FlashCopy with 64 KB grains	Higher	Faster	Slower	Lower
FlashCopy with 256 KB grains	Lower	Slower	Faster	Higher

Grain size: It is advisable to use the default 256 KB grain size for FlashCopy in most situations.

2.9 Planning for supportability

There are a range of choices and support options when you set up replication services in an SVC or Storwize V7000 environment. This chapter briefly covers several tips and tools that might be useful in ensuring that the configuration you install will be usable and supportable:

- Understanding consistency
- Planning for host attachments
- Extended Quorum support for SVC Stretched Cluster
- Setting up Support notifications

One of the more complex areas to plan can be the SAN fabric, especially when using long distance connections with WDMs or FCIP routers. For more information about planning your fabric, refer to Chapter 12, “Fabric design considerations for Replication Family” on page 473.

2.9.1 Understanding consistency

When copying the contents of one volume to another, for the copy to be of value, it must contain a write-order consistent set of data, sometimes referred to as crash-consistent. That is, the data set is in the same state it would be in if the power had failed and the system had come to a sudden unplanned stop. File systems and databases are designed to be able to cope with their logical disks being in such a state, and run recovery routines such as fsck and replaying redo logs.

SVC and Storwize V7000 are designed to always maintain write order consistency in their operations, with the exception of background copying. When synchronizing two volumes either initially or after a disruption, the background copying catch-up process is carried out without regard to the original write order. Until the volumes are synchronized, there is no consistency in the target volume. After they are synchronized, then ongoing updates are write-order consistent.

Functions such as Metro and Global Mirror and FlashCopy also support *Consistency Groups* whereby multiple volumes will also be maintained with write-order-consistency across the group. This ensures, for example, that a database volume does not fall out of write-order with a log file volume.

Although the system can guarantee write-order consistency, layers of write cache above the SVC or Storwize V7000 can prevent the system from understanding application consistency. Fortunately, applications are designed to cope with consistency recovery, for example after a power failure.

Where you want to maintain end-to-end application consistency, this can be provided in some situations by Tivoli Storage FlashCopy Manager.

How write cache affects consistency

The first computer to implement cache was the IBM System/360 Model 85, which was released in 1968. Since then, caches have been used to improve performance. Modern computer systems have multiple layers of cache, which means that an application cannot tell the difference between writes that are committed immediately to disk and writes that are cached awaiting later destage to disk. This is not unique to SVC or Storwize V7000.

Figure 2-8 shows that cached writes are not immediately committed to disk.

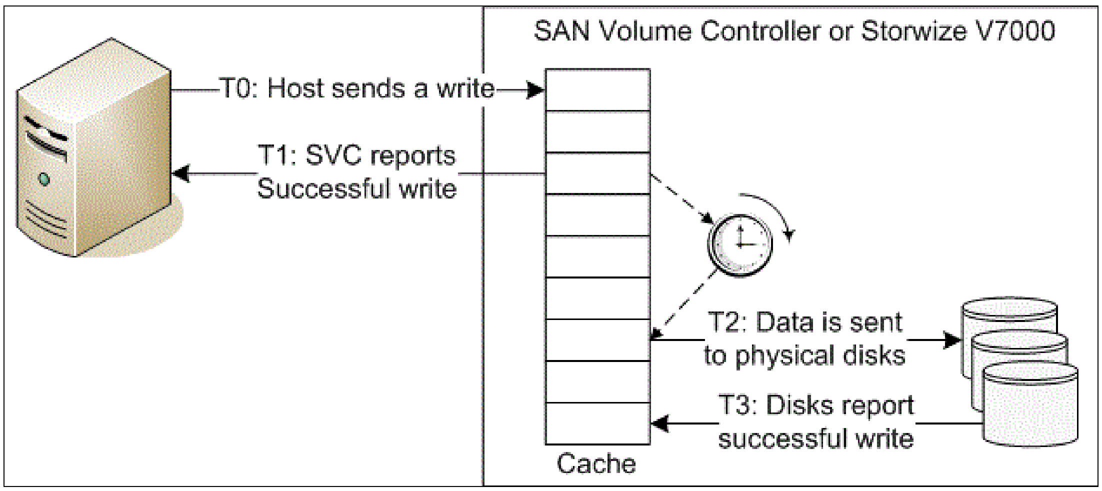


Figure 2-8 Cached writes are not immediately committed to disk

Transferring data to a write cache can be much faster than committing the data directly to disks. This write to cache occurs between T0 and T1 as shown in Figure 2-8. At T2, the write is destaged from cache to the physical disk, which completes at time T3.

Figure 2-9 on page 29 further illustrates that there can be many caches between the application and the disks.

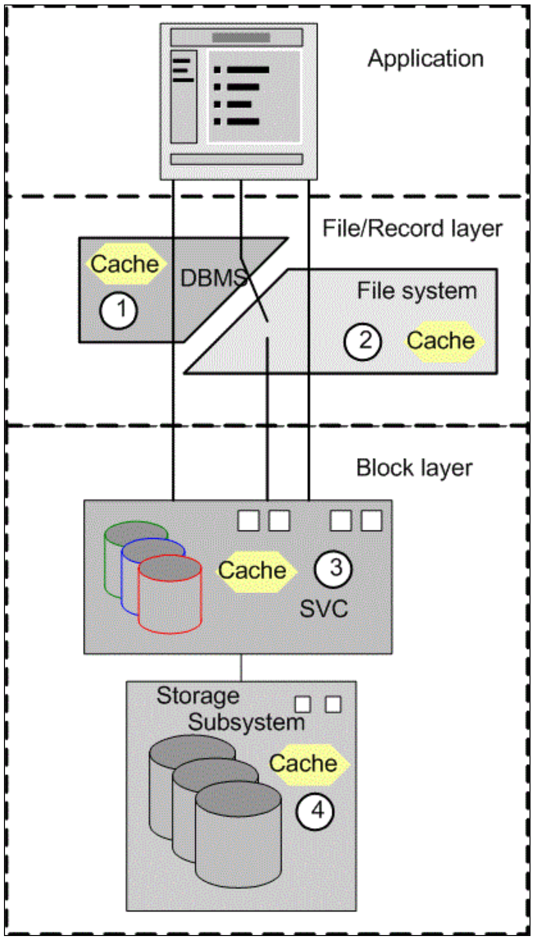


Figure 2-9 Layers of caching in a typical system

Destage timing: Each cache makes a commitment to the layer above that any data that it accepts will eventually be destaged. However, there is no commitment as to *when* the data will be destaged. The timing of the destage is the issue you need to provide for when planning a solution that uses copy services.

Consider the FlashCopy function, which provides a point-in-time copy of your data. Timing becomes important when creating a point-in-time copy, and yet we know that data written at t_0 might not actually be committed to the volume until t_1.

To ensure that the FlashCopy target volume contains a copy that can be used later, the host system must be configured to flush all the upper layer caches to disk and freeze the I/O during the start of the FlashCopy task. You configure this situation by:

- Quiescing I/O operations (for example, stopping applications)
- Setting caches to a write-through mode
- Setting applications to special backup modes that allow for consistent copies to be made

Tivoli Storage FlashCopy Manager: The easiest way to ensure application consistency with FlashCopy is to use Tivoli Storage FlashCopy Manager.

The same situation applies to Metro Mirror or any other form of replication. The primary and secondary volumes remain in sync with one another, but there can be a lag between the volumes and the application itself. When the relationship is broken or stopped, the secondary volume does not completely match the data expected by the application. To use this secondary volume, the application must make use of logging features to roll back the required transactions exactly as though recovering after a power failure.

2.9.2 Planning for host attachments

Host attachment protocols do not usually directly affect Replication Family services, other than the generic interactions which relate to general volume access on SVC and Storwize V7000. However, there are certain cases where the host operating system and the protocol choice is important.

For example, the new **`movevdisk`** command (used to move a volume non-disruptively between I/O groups) is dependent on predictable intelligent behavior from the multipath driver. Therefore, it will initially have limited support until thorough testing can be completed (for example, unclustered, Red Hat 6.x, SUSE Linux Enterprise Server 11, and possibly Microsoft Windows 2008, attached through a FC or FCoE).

The **`movedisk`** command is not currently supported with iSCSI and not supported with clustered hosts. Some clusters, for example, VMware, require that the SCSI LUN ID does not change on a volume. At the time of writing, **`movevdisk`** cannot force the SCSI LUN ID to stay the same after a move. Check the latest documentation for possible updates to this functionality.

Supported host configurations are documented in the following places:

- System Storage Interoperation Center (SSIC)
  `http://www.ibm.com/systems/support/storage/ssic`
- SVC interoperability pages
  `http://www.ibm.com/systems/storage/software/virtualization/svc/interop.html`
- Storwize V7000 interoperability pages
  `http://www.ibm.com/systems/storage/disk/storwize_v7000/interop.html`

Using the System Storage Interoperation Center

Although the SSIC is supposed to be the definitive guide on interoperability, it can be less informative, less current, and somewhat more difficult to use than the interoperability pages.

When checking for interoperability on an SVC, select **IBM System Storage SAN Volume Controller** in the first panel, and then choose your area of interest:

- SVC Host Attachment
- SVC Inter Cluster SAN Router Support
- SVC Storage Controller Support

Figure 2-10 on page 31 shows the initial selection panels for SVC in the SSIC.

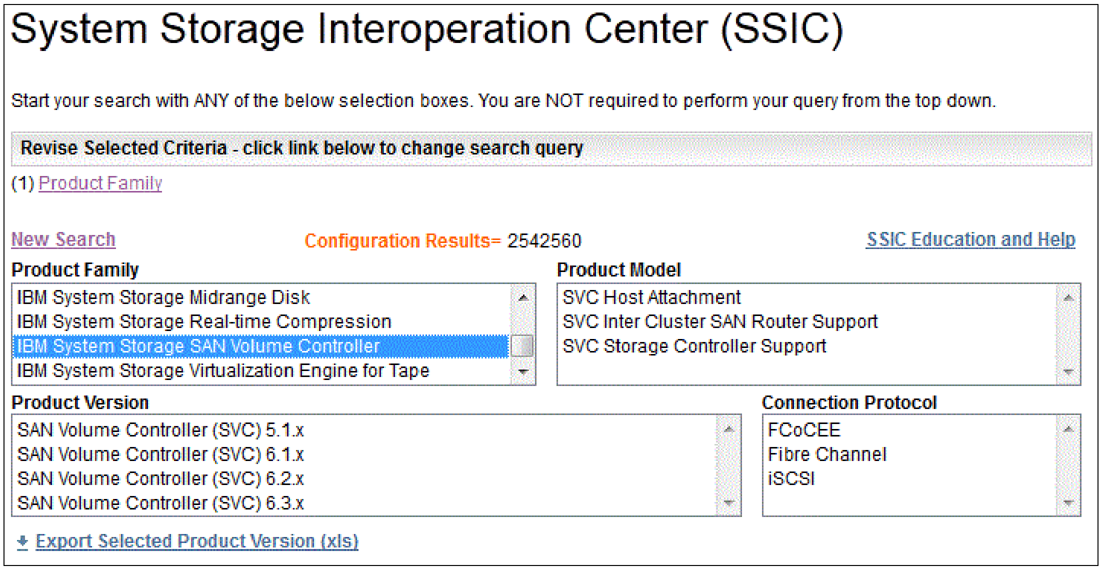


Figure 2-10 SVC interoperability in the SSIC

When checking for interoperability on a Storwize V7000, select **IBM System Storage Midrange Disk** in the first panel, and then choose your area of interest:

- Storwize Host Attachment
- Storwize Inter Cluster SAN Router Support
- Storwize Storage Controller Support

Figure 2-11 shows the initial selection panels for Storwize V7000 in the SSIC.

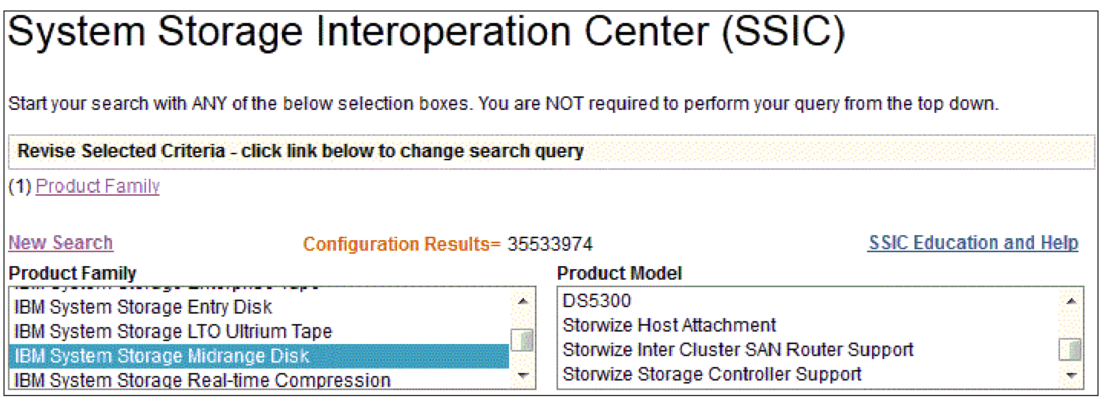


Figure 2-11 Storwize V7000 interoperability in the SSIC

As you complete the panels, the number of valid configurations will get smaller. After you have your subset fully defined the SSIC will give you the option of exporting the list, with appropriate notes and caveats, to an xls format file.

Interoperability pages: Although the SSIC is supposed to be the definitive guide on interoperability, also use the interoperability pages, which are more flexible and easier to navigate, and sometimes more current.

Non-standard hosts

If the host combination you want to implement does not appear on the SSIC or on the interoperability pages, the configuration might still be supported under special conditions. Sometimes it is simply a question of a particular combination not having been tested yet. If you contact the IBM account team and document what you need, the team might agree to submit a Storage Customer Opportunity REquest (SCORE) for support of a specific well-defined combination. This is also sometimes referred to as a Request for Price Quotation (RPQ).

iSCSI hosts

iSCSI support was introduced with code level 5.1. It has undergone significant advances with functionality improvements in 6.2 (CPU overhead reduction and 10 Gbps support) and in multi-session support in 6.3.0.1. SVC and Storwize V7000 iSCSI now provide high performance connectivity with up to 1150 MBps on a 10 Gbps link.

With the introduction of 10 Gbps support for iSCSI in code level 6.2, we no longer process each packet of traffic separately, but instead process pages or bundles (for example, 50 at a time).

To achieve the best performance from this approach it is important to turn off the "Delayed ACK" setting on the host operating system. "Delayed ACK" is the default setting for iSCSI on Windows and VMware hosts, so you will need to manually turn this setting off. Leaving "Delayed ACK" on will slow down reads from an SVC or Storwize V7000 system running at code level 6.2 or later.

In VMware ESX/ESXi 4.x/5.x, you can configure the `delayed_ack` setting per discovery target, per individual target, or globally for all iSCSI targets. The parameter setting is on or off. The default is 1 (on), so you will need to set `delayed_ack=0` (off).

In MS Windows, the setting describes how many outstanding pages there can be before an ACK is sent. You will need to set the registry entry for `TcpAckFrequency` to 1 (the default is 2).

> **Disabling "Delayed ACK" on the host operating system:** As of version 6.2, SVC and Storwize V7000 iSCSI provide high performance 10 Gbps connectivity, but it is essential to disable "Delayed ACK" on the host operating system.

See also "Guidelines for the Attachment of VMware iSCSI Hosts to SAN Volume Controller and Storwize V7000 - Errata, Version 6.2.x and Higher," which is available at the following site:

ftp://ftp.software.ibm.com/storage/san/sanvc/V6.3.0/VMware_iSCSI_Host_Attach_errata_V3.pdf

2.9.3 Extended Quorum Support for SVC Stretched Cluster

Every SVC has three quorum mdisks that can be used to arbitrate in the event of a cluster "split brain" condition. SVC Stretched Cluster has a specific requirement for the primary quorum to be on a disk system that is listed as supporting Extended Quorum on the SVC interoperability web site at:

http://www-03.ibm.com/systems/storage/software/virtualization/svc/interop.html

It is important to realize that *quorum* disk support and *Extended Quorum* disk support are two separate things. Only a small subset of the supported disk system types support Extended Quorum.

The current supported disk systems for Extended Quorum are:

- Storwize V7000 (requires SVC 6.2.0.3 or later)
- DS3400, IBM DS4000® family, DS5000 family
- ESS, DS6800, IBM DS8000® family

Figure 2-12 shows an example from the website of a disk system listed as supporting Extended Quorum for SVC Stretched Cluster.

Controller Model	Firmware	Supports SAN Volume Controller Quorum Disks	Support for MDisks >2TB	Multipathing	Notes
IBM Storwize V7000	Version 6.1.0.2 and Higher	Yes (Also supports Extended Quorum with SVC level 6.2.0.3)	Yes	Round Robin	Please visit the SAN Volume Controller Information Center for details on Configuring and Servicing external storage systems Please visit the following flash for details on Storwize V7000 as Backend Storage for SAN Volume Controller

Figure 2-12 Extended Quorum Support

Note that a rare few disks systems (most notably HP MSA, and HDS 99xx including its OEM variants) do not support quorum or extended quorum disks at all. With SVC Stretched Cluster, we place each of the three quorum MDisks on disk systems that are in different power domains to try to ensure that at least one quorum MDisk will always be available. After your quorum MDisks are established you can use the **`lscontrollerdependentvdisks`** command to validate that the quorum MDisks do indeed reside on separately located disk systems.

2.9.4 Support Notifications

Support Notifications is a free service. It is strongly advised that all owners of IBM SVC and Storwize V7000 use this service. To register for automated Support Notifications and alerts for SVC and Storwize V7000, go to:

`https://www.ibm.com/support/entry/myportal/overview`

In the center of the page is a panel labelled “Notifications.” Click **Sign in to create, manage or view your subscriptions**. After you are registered and logged in, select the **Subscribe** tab and then select either **Storage Software** → **SAN Volume Controller or select Disk**

Systems → **IBM Storwize V7000**. Figure 2-13 on page 34 shows setting up a “My notifications” subscription.

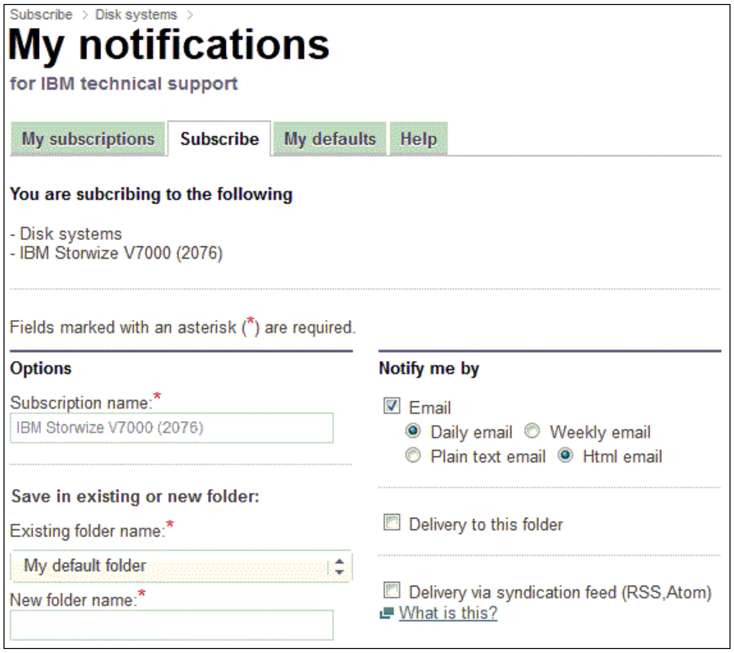


Figure 2-13 My notifications

3

Metro Mirror and Global Mirror

This chapter discusses the features of Metro Mirror and Global Mirror.

- Metro Mirror is the IBM name for synchronous volume replication.
- Global Mirror is the IBM name for continuous asynchronous volume replication.
- Global Mirror with Change Volumes is the IBM name for point-in-time asynchronous volume replication. It is a new option introduced with Version 6.3.

Also new in Version 6.3 is support for Metro and Global Mirror between a SAN Volume Controller (SVC) and Storwize V7000.

All of the features mentioned in this chapter apply equally to any combination of SVC and Storwize V7000, unless otherwise specified.

3.1 Terminology

In describing Metro and Global Mirror, the following terminology applies:

I/O group An I/O group is a pair of SVC nodes or Storwize V7000 node canisters. An I/O group can be thought of as an active/active pair of controllers.

Clustered system A clustered system consists of one, two, three or four I/O groups. Historically referred to as a *cluster*, and now more commonly as a *system*.

Partnership A Metro or Global Mirror partnership is a pairing of two three or four SVC or Storwize V7000 systems. Each system can maintain up to three partnerships, and each partnership can be with a single remote system. Note that in some cases you can create partnerships between systems with different software levels.

Volume A volume is broadly synonymous with a logical disk and a LUN and was referred to as a VDisk in the past.

Consistency Group A Metro or Global Mirror consistency group is a structure into which can be added zero or more volumes, typically to ensure application consistency across a set of volumes, or simply for ease of management. Note that at the time of writing, VMware Site Recovery Manager did not support the use of Metro or Global Mirror consistency groups.

Relationship A Metro or Global Mirror relationship is a pairing of two volumes or consistency groups. When a relationship is first created, one side is labelled as the *master* and the other as the *auxiliary*. Either master or auxiliary can, however, be assigned the roles of *primary* (source) or *secondary* (target) in the relationship.

3.2 Metro and Global Mirror overview

Metro Mirror is designed for metropolitan distances in conjunction with a zero Recovery Point Objective (RPO); that is, zero data loss. This is achieved with a synchronous copy of volumes; writes are not acknowledged until they are committed to both storage systems. By definition, any vendors' synchronous replication will make the host wait for write I/Os to complete at both the local and remote storage systems and will include round-trip network latencies. Metro Mirror has the following characteristics:

- Zero RPO
- Synchronous
- Production application performance impacted by round-trip latency

Global Mirror is designed to minimize application performance impact by replicating asynchronously, that is, writes are acknowledged as soon as they can be committed to the local storage system, sequence-tagged and passed on to the replication network. This allows Global Mirror to be used over longer distances. By definition, any vendors' asynchronous replication will result in an RPO greater than zero. However, in the case of Global Mirror the RPO is quite small, typically anywhere from several milliseconds to some number of seconds.

Although Global Mirror is asynchronous, the RPO is still quite small and thus the network and the remote storage system must both still be able to cope with peaks in traffic. Global Mirror has the following characteristics:

- Near-zero RPO
- Asynchronous
- Production application performance impacted by I/O sequencing preparation time

Global Mirror with Change Volumes provides an option to replicate point-in-time copies of volumes. This will generally require lower bandwidth because it is the average rather than the peak throughput that must be accommodated. The RPO for Global Mirror with Change Volumes will be higher than traditional Global Mirror. Global Mirror with Change Volumes has the following characteristics:

- Larger RPO
- Point-in-time copies
- Asynchronous
- Possible system performance overhead because point-in-time copies are created locally

While Metro Mirror or Global Mirror is active, the secondary volume is available in a read-only state. Accessing the secondary volume for read/write would first require that the Mirror relationship be interrupted.

Figure 3-1 shows how you can choose the type of relationship you want to create. This is located from the Overview panel → Copy Services → Remote Copy → New Relationship.

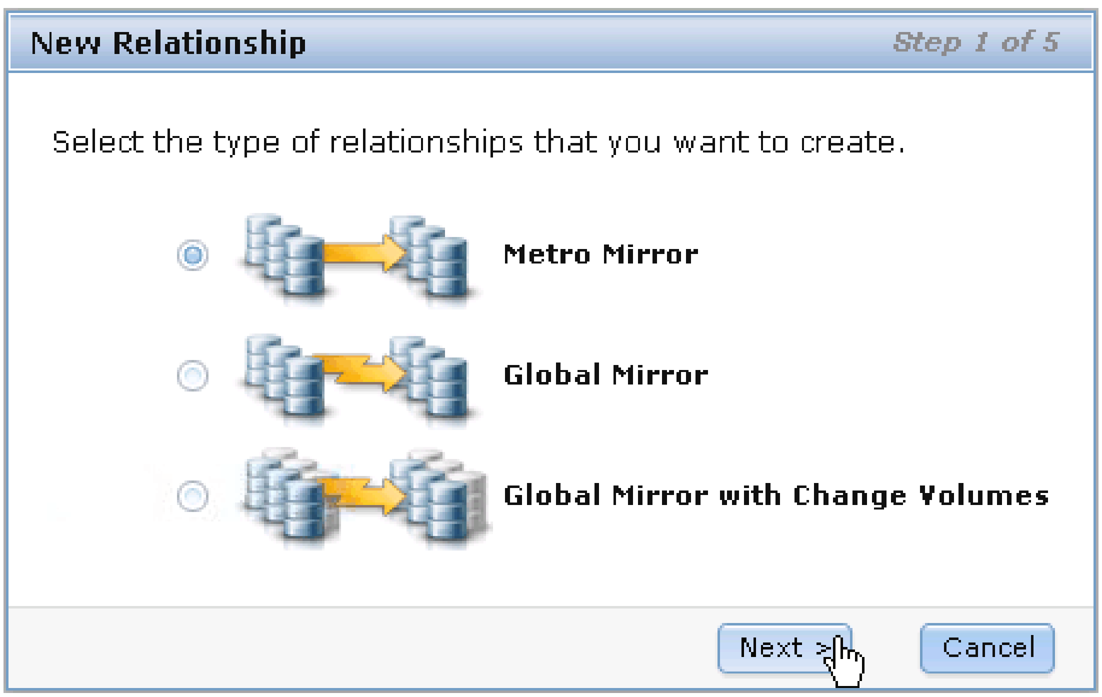


Figure 3-1 Choose the type of relationship you want to create

3.3 Partnerships between systems

When creating a Global or Metro Mirror partnership, you establish a control link between two systems. The control link between the systems uses the same FC or FCoE paths as the data that will be replicated.

Partnerships are established between two systems by issuing the **`mkpartnership`** command once from each end of the partnership. The only parameters that need to be specified are the remote cluster name (or ID) and the maximum background copy bandwidth in megabytes per second (MBps) you want available in that particular direction. The bandwidth parameter will determine the maximum speed of the initial synchronization of the relationships.

Figure 3-2 shows creating a partnership from the GUI.

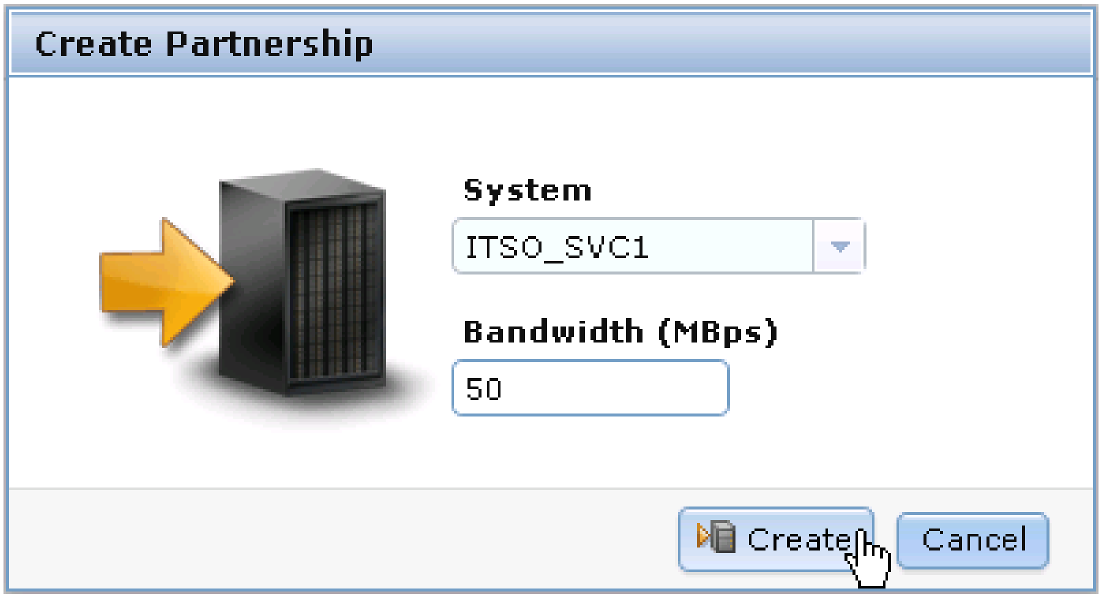


Figure 3-2 Creating a partnership with the GUI

Partnership consideration: When creating the partnership, there is no master/auxiliary status defined or implied. The partnership is equal and the concepts of master/auxiliary and primary/secondary only apply to volume relationships, not to system partnerships.

Metro Mirror and Global Mirror relationships are designed to accommodate major events, such as failover, recovery, and catch-up resynchronization, although with a degree of skilled user intervention. Catch-up resynchronization is supported so that a relationship failure does not require a complete from-scratch resynchronization of the relationship.

For a higher degree of automation in failover and failback, it is advisable to consider adding Tivoli Storage Productivity Center for Replication (TPC-R) to your environment. See Chapter 11, “Software solutions and services based on IBM SVC Replication Family Services” on page 427 for more information about this topic.

Each system can maintain up to three partner cluster relationships, thereby allowing as many as four clusters to be directly associated with each other.

Figure 3-3 on page 39 shows an example of a system participating in multiple partnerships.

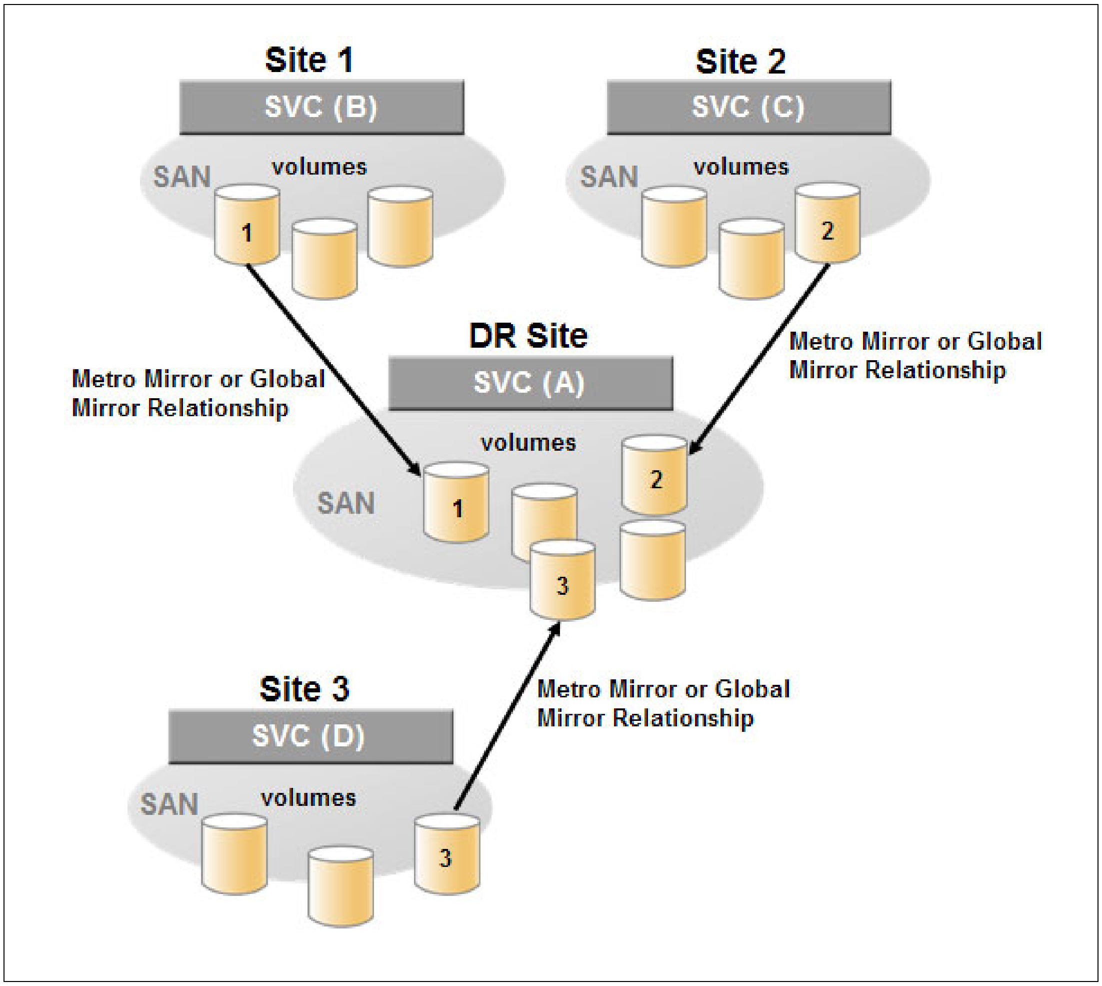


Figure 3-3 Each system can participate in up to three partnerships

Partnership restrictions: The following partnership restrictions apply:

- Systems running code level 6 or later cannot form a partnership with systems running code levels below 4.3.1. Partnerships between systems running code level 6 and code level 4.3.1 cannot participate in additional partnerships with other systems.
- Systems that are all running code level 5.1 or later will support up to three partnerships.
- Systems running code level 6 support object names up to 63 characters. However, when they are partnered with systems running code level 5.1 or lower, those lower version partners will display object names truncated at 15 characters.

3.3.1 Layer concept

Version 6.3 introduced the concept of *layers*. The key points concerning layers are listed here:

- SVC is always in *Replication* layer.
- By default, Storwize V7000 is in *Storage* layer.
- A system can only form partnerships with systems in the same layer.

- An SVC can virtualize a Storwize V7000 only if the Storwize V7000 is in Storage layer.
- With version 6.4, a Storwize V7000 in Replication layer can virtualize a Storwize V7000 in Storage layer.

Figure 3-4 illustrates the concept of layers.

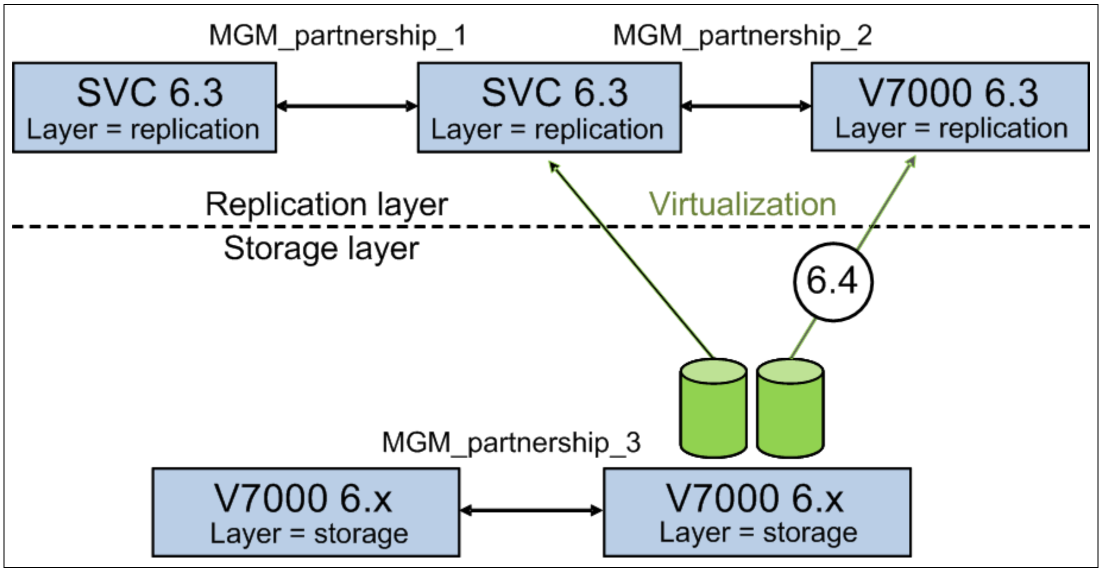


Figure 3-4 Conceptualization of layers

Generally, changing the layer is only performed at initial setup time or as part of a major reconfiguration. To change the layer of a Storwize V7000, the system must meet the following pre-conditions:

1. The Storwize V7000 must not have any SVC or Storwize host objects defined, and must not be virtualizing any other Storwize controllers.
2. The Storwize V7000 must not be visible to any other SVC or Storwize system in the SAN fabric, and this might require SAN zoning changes.
3. The Storwize V7000 must not have any system partnerships defined. If it is already using Metro Mirror or Global Mirror, the existing partnerships and relationships must be removed first.

Changing a Storwize V7000 from Storage layer to Replication layer can only be performed using the CLI. After you are certain that all of the pre-conditions have been met, issue the following commands:

```
chsystem -layer replication
lssystem
```

Storwize V7000 consideration: To partner with an SVC, a Storwize V7000 must be running 6.3 or later and be running in Replication layer.

3.3.2 Partnership topologies

Each system can be connected to a maximum of three other systems for the purposes of Metro or Global Mirror.

Figure 3-5 on page 41 shows examples of the principal supported topologies for Metro and Global Mirror partnerships. Each numbered box represents an SVC or Storwize V7000 system.

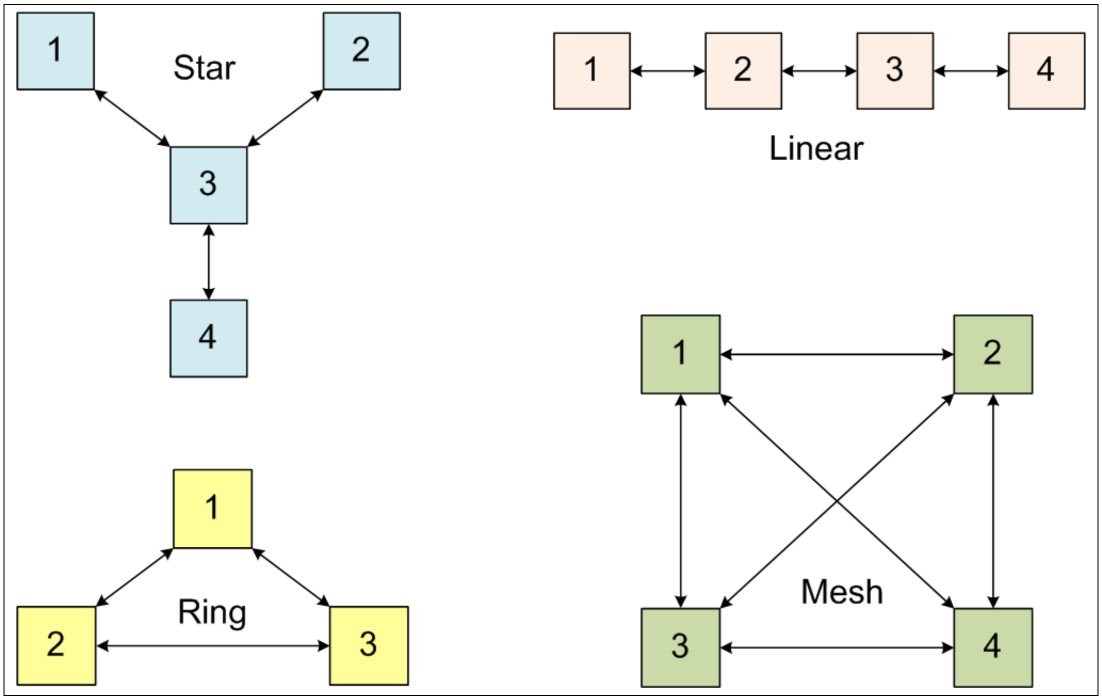


Figure 3-5 Supported topologies for Metro and Global Mirror

Star topology

A star topology can be used, for example, to share a centralized disaster recovery system (3, in our example) with up to three other systems, for example replicating 1 → 3, 2 → 3, and 4 → 3.

Ring topology

A ring topology (3 or more systems) can be used to establish a one-in, one-out implementation. for example, 1 → 2, 2 → 3, 3 → 1 so as to spread replication loads evenly among three systems.

Linear topology

A linear topology of two or more sites is also possible, although it would generally be simpler to create partnerships between system 1 and system 2, and separately between system 3 and system 4.

Mesh topology

A fully connected mesh topology is where every system has a partnership to each of the three other systems. This allows flexibility in that volumes can be replicated between any two systems.

Topology considerations:

- Although clusters can have up to three partnerships, any one volume can be part of only a single relationship. That is, you cannot replicate any given volume to multiple remote sites.
- Although a variety of topologies is supported, it is advisable to keep your partnerships as simple as possible, which in most cases will mean system pairs or a star.

3.3.3 Partnership heartbeat

All SVC nodes maintain a database of the other devices that are visible on the fabric. The database is updated as devices appear and disappear. Nodes that belong to the same system establish communication channels among themselves and begin to exchange messages to implement the clustering functions. Nodes that are in different systems (clusters) do not exchange messages after the initial discovery is complete, unless they have been configured in a Metro or Global Mirror partnership.

When a partnership is established, a *heartbeat* message is sent between systems to check the connection. The node from each system that owns the heartbeat is termed the *focal point.* The traffic between the focal point nodes is distributed among the logins that exist between those nodes. If the focal point node fails, or all its logins to the remote cluster fail, a new focal point is chosen to run the control traffic.

The software that controls the inter-cluster link monitors for link errors and excludes the link if the error rate gets too high. The events that can eventually lead to an excluded link are:

- A hardware problem (link failure) on a non-redundant link between the two systems
- An intermittent hardware problem on a non-redundant link that leads to a loss of credit over time that leads to a link reset
- Lost frames or a delay exceeding 1.5 seconds that affects the heartbeats between the two clusters
- A problem leading to excessive data frames being dropped (even where non-data frames are passed successfully)
- A series of failures that means that progress is not made, such that a message is not delivered for longer than 15 seconds

Large but short-term disruption is tolerated, but if within a 10-minute period there are more than three 30-second periods that experience an error event, then the link is excluded.

When a link is excluded, the other cluster will report `partially_configured` in its configuration state for the remote cluster. It is likely the other cluster will also have detected the same set of errors and excluded the link on the same basis, in which case both clusters will report `partially_configured`. If only one cluster detects an error and excludes the link, that cluster will continue to report `fully_configured`.

3.4 Network connectivity for Metro and Global Mirror

Metro and Global Mirror are designed to work with Fibre Channel (FC). Standard FC distance limits apply for shortwave multimode and longwave single mode cabling. For longer distances, Fibre Channel over Internet Protocol (FCIP) routers are supported, as are wave division multiplexors (WDMs).

A separate control network is not required for Metro or Global Mirror. The control link is implemented on top of the same Fibre Channel fabric connection that is used for the volume replication itself.

3.4.1 Fibre Channel over Ethernet (FCoE)

With Version 6.4, Fibre Channel over Ethernet (FCoE) is now supported for both host connectivity and Metro and Global Mirror, where 10 Gbps Ethernet ports are available. Configuration of Metro or Global Mirror over FCoE is similar to configuration over Fibre Channel.

Figure 3-6 shows that FCoE ports look just like FC ports to the **lsfabric** style view in the GUI.

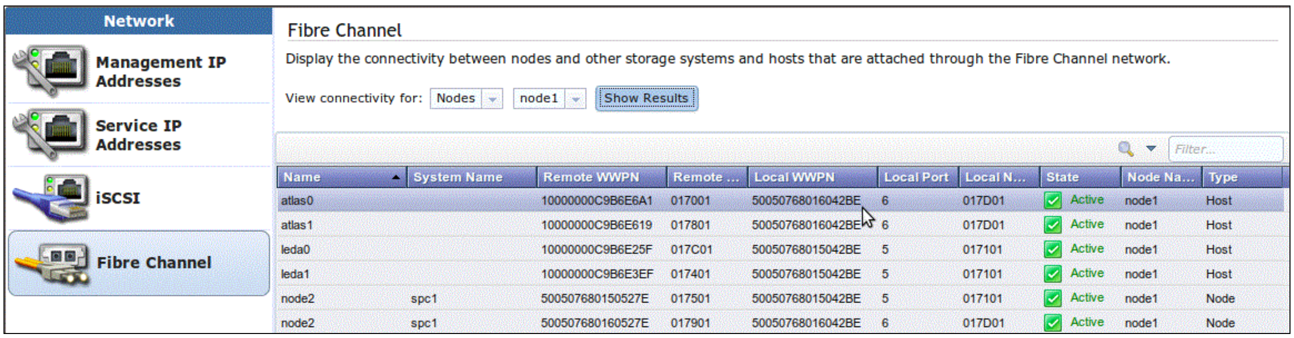


Figure 3-6 FCoE ports

Figure 3-7 shows part of the canister hardware view in the GUI with FCoE ports active and FC ports inactive.

Ports

WWPN	Status	Speed	Type
50050768016042BE	Active	10Gb	Ethernet
50050768015042BE	Active	10Gb	Ethernet
50050768014042BE	Inactive	8Gb	Fibre Channel
50050768013042BE	Inactive	8Gb	Fibre Channel
50050768011042BE	Inactive	8Gb	Fibre Channel
50050768012042BE	Inactive	8Gb	Fibre Channel

Figure 3-7 Canister hardware view showing active FCoE ports and inactive FC ports

Avoiding protocol confusion

Sometimes the concepts of Ethernet and IP are confused, because the two have traditionally been so closely associated. *Ethernet* is a data link layer technology (Layer 2) and *Internet*

Protocol (IP) is a network/internet layer protocol (Layer3). By extension, sometimes the concepts of FCoE and FCIP are also confused.

FC frames contain SCSI commands and data for transmission, generally over fiber optic cables. FC typically runs over shorter distances, for example, 150 m with multimode fiber, depending on the optical medium and the transceivers selected. Or FC runs over longer distances, for example 4 km, 10 km, or 25 km with single mode fiber, depending on the transceivers used. FC and its associated technology is feature-rich, highly reliable, and generally regarded quite stable.

FCoE encapsulates FC packets (which contain SCSI commands and data) inside Ethernet frames for use over data center class 10 Gbps Ethernet networks, with distance limitations similar to standard FC cabling. Currently the FCoE standard (T11 Backbone 5) uses a pause/go method of congestion control that is less sophisticated than the buffer credit system used by native FC. FCoE does, however, allow you to use converged network adapters (CNAs) to consolidate host-side cabling into a single link, rather than requiring separate links for IP and FC traffic.

Note that the current FCoE standard (BB-5) cannot route across fabrics. When you need to route across fabrics, you therefore must include FC or FCIP devices in your network design. To date, FCoE has mostly been used between the top-of-rack switches and the hosts, rather than as the core storage network.

Figure 3-8 shows that the FC-0 and FC-1 layers are replaced in FCoE.

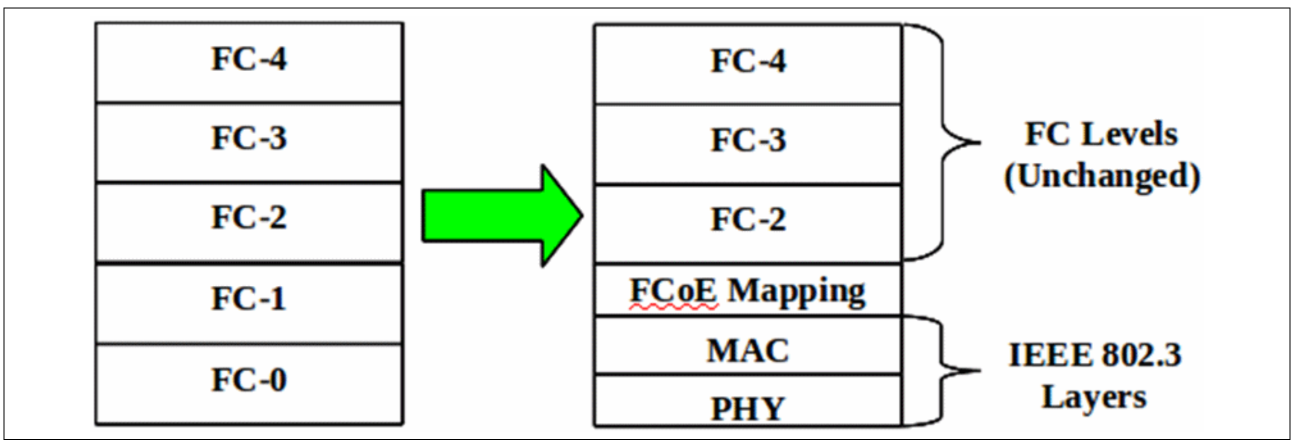


Figure 3-8 FC-0 and FC-1 are replaced in FCoE

Figure 3-9 shows how the FC packet is encapsulated inside an Ethernet frame.

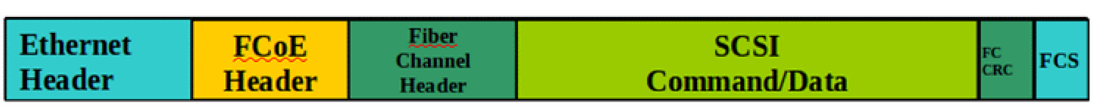


Figure 3-9 FCoE encapsulates FC packets

FCIP is a tunnelling protocol that is designed to take FC frames, which contain SCSI commands and data, and wrap them in IP packets. The IP packets can then be transmitted across long distance links which might have variable reliability. This is designed to overcome the distance limitations of Fibre Channel.

Unlike these other technologies, *iSCSI* is not an augmentation of FC but an alternative to it. iSCSI is supported for host attachment, but not for Metro or Global Mirror. A more detailed discussion of iSCSI is therefore beyond the scope of this chapter.

> **iSCSI consideration:** Although iSCSI is supported for host attach, it is not supported for Metro or Global Mirror replication.

Figure 3-10 on page 45 shows the ports on an SVC 2145-CG8 node that can be used for Metro or Global Mirror traffic.

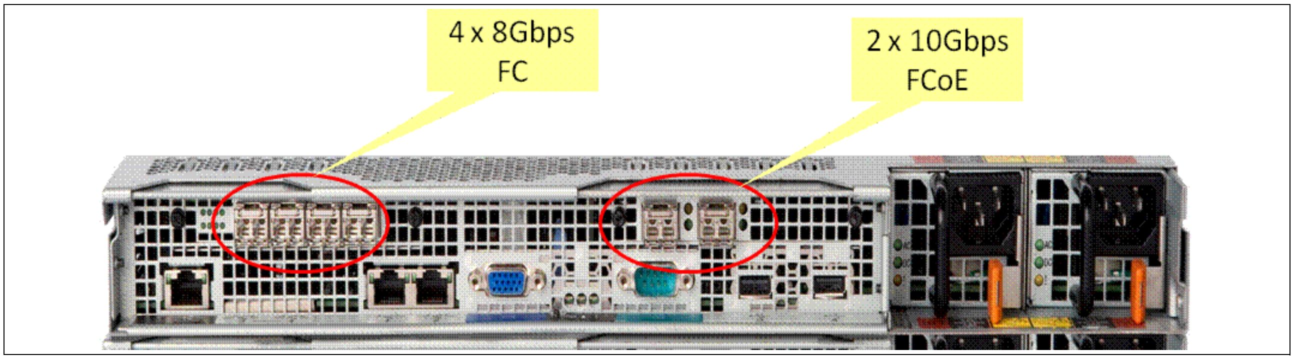


Figure 3-10 FC and FCoE ports on an SVC 2145-CG8 node

3.4.2 Planning your network for Metro or Global Mirror

It is important to consult the latest *Software Installation and Configuration Guide* when planning a Metro or Global Mirror network because it outlines key requirements for support:

http://publib.boulder.ibm.com/infocenter/svc/ic/topic/com.ibm.storage.svc.console.doc/svc_bkmap_confguidebk.pdf

Inter-system bandwidth should be capable of supporting the combined traffic of the following items:

- Mirrored foreground writes, as generated by your server applications at peak times
- Background re-synchronization, for example, after a link outage
- Inter-system heartbeat

Calculating the required bandwidth is essentially a question of mathematics based on your current workloads, so it is advisable to start by assessing your current workloads.

For Metro or Global Mirror, you will need to know your peak write rates and I/O sizes down to at least a 5-minute interval. This information can be easily gained from Tivoli Storage Productivity Center for Disk (TPC for Disk), which is an IBM licensed product.

There are also unsupported tools available from IBM:

http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/TD105947

https://www.ibm.com/developerworks/mydeveloperworks/blogs/svcmon

Network sizing will also need to provide for heartbeat traffic of between 2.6 Mbps for a 2-node to 2-node replication, and 12.4 Mbps for an 8-node to 8-node replication. Note that Metro and Global Mirror bandwidth parameters are set in MBps but heartbeat overheads are quoted in Mbps.

> **Workload data:** Effective bandwidth sizing requires that you start with accurate data about your workload.

We strongly advise that you do not compromise on bandwidth or network quality when planning a Metro or Global Mirror deployment. If bandwidth is likely to be an issue in your environment, we suggest you consider Global Mirror with Change Volumes as described in 3.11.2, “Bandwidth sizing example for Change Volumes” on page 68.

In 3.8.2, “Partnership bandwidth (total background copy limit)” on page 57 we discuss bandwidth allocation for background copy rates. The concepts of network bandwidth for normal replication and partnership bandwidth for background copying are related. An estimate is to make an allowance of 20 percent for background copying for resynchronization after an outage.

Finally, you will need to allow for unexpected peaks (and any unrecorded peaks hidden by TPC’s 5-minute averaging) plus planned growth.

When deploying Global Mirror across an FCIP link using IBM SAN06B-R (2498-R06) FCIP routers, it is recommended that they be licensed with the *Integrated Routing* feature code 7739 to provide fabric isolation across the link. This is mandatory for Stretched Cluster environments using FCIP. For more information about network connectivity, refer to Chapter 12, “Fabric design considerations for Replication Family” on page 473.

3.5 Relationships between volumes or Consistency Groups

A Global or Metro Mirror relationship is the connection between two volumes or consistency groups.

When a relationship is first created, the master volume is always assigned the role of the primary, and the auxiliary volume is assigned the role of the secondary. These roles can be reversed at any stage, with the auxiliary becoming the primary and the master becoming the secondary.

Figure 3-11 shows that primary and secondary roles are distinct from the concept of master and auxiliary volumes.

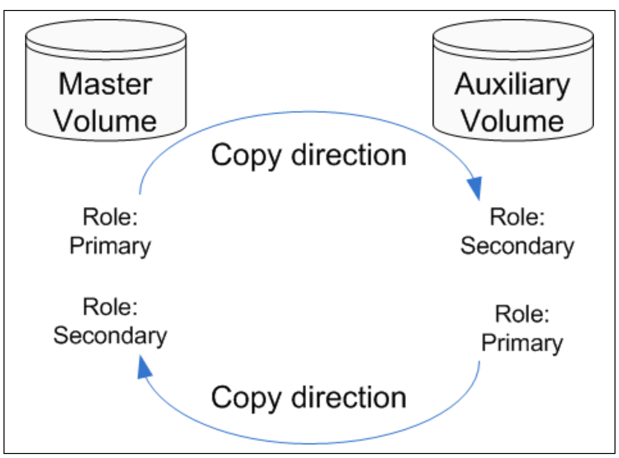


Figure 3-11 Master and Auxiliary volumes can each take the primary or the secondary role

When you create a copy relationship, you need to state which volume is the master. After it is set, this designation does not change even if you change copy directions.

Example 3-1 shows the creation of a relationship specifying which volume is the master and which the auxiliary.

Example 3-1 Master and auxiliary are assigned when the relationship is created

```
mkrcrelationship -master master_vol -aux aux_vol -cluster remote_system
```

You can also optionally specify whether the relationship should be added to a Consistency Group and whether it should be a Global Mirror relationship, or a Global Mirror with Change Volumes relationship, or a Metro Mirror relationship, which is the default.

The **mkrcrelationship** command also allows you to define whether the relationship will initially be synchronized through background copying or by an offline method such as tape.

3.5.1 Consistency Groups

A Consistency Group (CG) is a collection of relationships that can be treated as one entity, usually to preserve write order consistency across a group of volumes that pertain to one application, for example, a database volume and a database log file volume.

After a volume relationship is added into a Consistency Group, you cannot manage the relationship in isolation from the Consistency Group. So, for example, issuing a **stoprcrelationship** on the stand-alone volume would fail because the system knows that the relationship is part of a Consistency Group.

Note the following points regarding Consistency Groups:

- Each volume relationship can belong to only one Consistency Group.
- Volume relationships can also be stand-alone, that is, not in any Consistency Group.
- Consistency Groups can also be created and left empty, or can contain one or many relationships.
- You can create up to 256 Consistency Groups on a cluster (system).
- All volume relationships in a Consistency Group must have matching primary and secondary clusters (systems), but they do not need to share the same I/O groups.
- All relationships in a Consistency Group will also have the same copy direction and state.
- Each Consistency Group is either for Metro Mirror or for Global Mirror relationships, but not both. This is determined by the first volume relationship that is added to the Consistency Group.

Consistency Group consideration: A Consistency Group relationship does not have to be in a directly matching I/O group number at each site. A Consistency Group owned by I/O group 1 at the local site does not have to be owned by I/O group 1 at the remote site. If you have more than one I/O group at either site, you can create the relationship between any two I/O groups and thereby spread the workload, for example, from local I/O group 1 to remote I/O group 2.

Streams

Consistency Groups can also be used as a way to spread replication workload across multiple streams within a partnership.

The Metro or Global Mirror partnership architecture will allocate traffic from each given Consistency Group in a round-robin fashion across 16 streams. That is, cg0 traffic will go into stream0 and cg1 traffic will go into stream1.

Any volume that is *not* in a Consistency Group will also go into stream0. You might want to consider creating an empty Consistency Group 0 so that stand-alone volumes do not share a stream with active Consistency Group volumes.

It can also pay to optimize your streams by creating more Consistency Groups. Within each stream, each batch of writes must be processed in tag sequence order and any delays in processing any particular write will also delay the writes behind it in the stream. Having more streams (up to 16) reduces this kind of potential congestion. Also, each stream is sequence-tag-processed by one node, so generally you would want to create at least as many Consistency Groups as you have SVC nodes/Storwize V7000-canisters, and ideally, perfect multiples of the node count.

3.6 Initial synchronization

In 3.5, "Relationships between volumes or Consistency Groups" on page 46, the **`mkrcrelationship`** command in the context of defining a master and an auxiliary is discussed. Beyond this, **`mkrcrelationship`** and **`startrcrelationship`** also give you options for carrying out the initial synchronization.

3.6.1 Default synchronization across the network

By default, the system will copy the data from the primary to the secondary, and the background copy bandwidth rate will apply. If your network resources are limited, it is best not to set the background copy rate too high because it needs to share the available bandwidth with other active relationships. This caution does not apply to Global Mirror with Change Volumes.

See 3.8.2, "Partnership bandwidth (total background copy limit)" on page 57 and 3.8.3, "relationshipbandwidthlimit" on page 58, for more information about this topic.

The sequence for a single relationship is:

1. **`mkrcrelationship`** is issued without specifying the **`-sync`** flag.
2. **`startrcrelationship`** is issued without the **`-clean`** flag.

3.6.2 Tape-based synchronization

This method has a benefit over the default synchronization in that it does not require all the volume data to be copied over a constrained link. If you have many terabytes of data and a relatively small network, it might take a long time to complete an initial synchronization across the network, even though your network might be perfectly well-sized to handle the ongoing update traffic after the initial synchronization is complete.

There are two parameters of particular interest when considering tape-based synchronization, **`mkrcrelationship -sync`** and **`startrcrelationship -clean`**, as explained here.

mkrcrelationship -sync

This parameter tells the system that you are confident that the primary and the secondary are already synchronized (or will be before you issue a **`startrcrelationship -clean`**) so Global Mirror or Metro Mirror does not need to perform an initial sync. Often we use **`-sync`** to spoof a synchronized state to the system, knowing that we are in the process of manually creating the synchronized state we need.

startrcrelationship -clean

This parameter tells the system to ignore any writes that have occurred on the secondary, for example, since a **stoprcrelationship -access** was issued. The writes will have been journaled, but **-clean** tells the system to ignore what is in that journal. You can use this, for example, to load a backup tape image at the secondary site, without the system seeing this write activity as putting the secondary site into an inconsistent state compared to the primary. This is good because we know the reverse is true; that is, the writes coming from tape onto the secondary volume is the mechanism by which we are creating a consistent state.

Tape Synchronization with Writes Paused at the Primary

In most cases we would recommend the use of FlashCopy see "Tape synchronization with writes continuing at the primary" so as to minimize the pause time on the production volumes, but tape synchronization without using FlashCopy could also be useful for sites that can afford an outage on the production volumes, for example, perhaps over a weekend.

The following list describes the general approach you would take to achieve a tape-based synchronization, assuming you can afford to suspend access to the production volumes while the tape copies complete.

1. All normal system access is stopped.
2. An **mkrcrelationship** is issued with the **-sync** flag.
3. No writes are made to the primary; that is, the system is unavailable for normal use.
4. A complete image backup is taken of the primary, for example, using tape.
5. The image backup is restored onto the secondary, for example, from tape.
6. A **startrcrelationship** is issued with the **-clean** flag.

If these steps are not performed correctly, the relationship could be reported as being consistent when it is not consistent, which would render any secondary volume useless. In a worst case scenario an inconsistent secondary could go undetected until a disaster strikes.

Tape synchronization with writes continuing at the primary

You can use FlashCopy to create a point-in-time copy of a volume and thus minimize the production outage. In this method, the administrator must still create an image backup and restore it at the secondary site, but the production site can continue processing as soon as the FlashCopy is done, so this is quite a brief suspension of normal access. The steps are as follows:

1. All normal system access is stopped.
2. **mkrcrelationship** is issued with the **-sync** flag.
3. **stoprcrelationship** is issued with the -**access** flag. The system will begin journalling writes at both ends of the relationship.
4. A FlashCopy is taken.
5. Normal access is allowed to the production volumes.
6. A complete image backup is taken of the primary FlashCopy, for example, using tape.
7. The image backup is restored onto the secondary, for example, from tape.
8. **startrcrelationship** is issued with the **-clean** flag and the secondary will be updated with the volume writes that have happened since the **-access** was issued. The **-clean** flag

tells the system to ignore any earlier writes that have been journaled at the secondary site, for example, the writes from tape, because you know they only existed to create the synchronized image.

This is designed to allow processing to continue at the primary while you get your image backup completed, get your tapes to the secondary site, and get them loaded. Only the data that changed since the FlashCopy was created is copied from the primary volume to the secondary volume across the network.

> **Attention:** Image backups are not the same as standard file-based backups. Tape synchronization will only work if your backup is an *exact* disk image of the FlashCopy volume.
>
> Be sure you understand the tape synchronization steps thoroughly, because `-sync` and `-clean` will override the system's ability to detect any inconsistency between the primary and secondary volumes.
>
> By using this method you are taking personal responsibility for establishing consistency between the two volumes.

3.7 Relationship states

Relationships for stand-alone volumes and Consistency Groups share a common state model. Any relationships in a Consistency Group will have the same state as the Consistency Group itself. In the following descriptions, the word "relationship" is used to imply both stand-alone volume relationships and Consistency Groups.

3.7.1 State definitions

This section defines the terminology used to describe relationship states.

Connected

When the two systems can communicate, they are *connected*. When they cannot communicate, they are *disconnected*. Most of the states discussed in 3.7.2, "State descriptions" on page 51 relate to systems that are connected.

Consistent

A relationship between a primary and a secondary is *consistent* if the data on the secondary matches the data that existed on the primary at a point in time. The point in time is referred to as the *recovery point*. Consistency does not necessarily mean that the data is up to date.

The secondary can enter a consistent but *unsynchronized* state, for example, due to a network outage. This is also the typical state of disaster recovery volumes that are part of a Global Mirror with Change Volumes relationship.

The secondary is sometimes referred to as crash-consistent. Because write order consistency is strictly enforced, any application designed to cope with unexpected power failure should be able to use the secondary volume and begin operation just as though it had been restarted after a hypothetical power failure.

An *inconsistent* state can arise, for example, after an extended network outage after the resync process begins. The primary writes are tracked during the outage, but the write order is not tracked. When communication is restored, resync grains might be transmitted out of order, so the secondary state is inconsistent until the resync is complete.

Two approaches can be used to cope with this situation:

- Take a FlashCopy of the consistent secondary before allowing the resync process to begin. In the event of a disaster before consistency is achieved again, the FlashCopy provides a consistent, though out-of-date, image.
- Accept the loss of consistency during resync, but ensure there is enough additional network bandwidth for the resync to complete in a timely fashion.

Synchronized

A copy that is consistent and up-to-date is described as *synchronized*. In a synchronized relationship, the primary and secondary volumes are different only in regions where writes are outstanding in ordered flight across the network.

3.7.2 State descriptions

This section provides you with detailed descriptions of the states.

Consistent Stopped

The secondary contains a consistent image, but it might be out of date with respect to the primary. This state can arise when a relationship is in Consistent Synchronized state and suffers an error that forces a consistency freeze. It can also arise when a relationship is created with a sync parameter set.

Normally, following an I/O error, subsequent write activity causes updates to the primary, and the secondary is no longer synchronized. In this case, to reestablish synchronization, consistency must be given up for a period. A **`start`** command with the **`-force`** parameter must be used to acknowledge giving up consistency, and the relationship transitions to inconsistent copying. Perform this step only after all outstanding errors are repaired.

In the unusual case where the primary and secondary are still synchronized (perhaps following a user stop, and no further write I/O was received), a **`start`** command takes the relationship to Consistent Synchronized. The **`-force`** parameter is not required. Also in this unusual case, a **`switch`** command is permitted, which moves the relationship to Consistent Synchronized and reverses the roles of the primary and secondary (switches the copy directions).

If the relationship becomes disconnected, the secondary side transitions to Consistent Disconnected. The primary side transitions to Idling Disconnected.

An entry into the status log is generated every time that a relationship enters the Consistent Stopped with a status of Online state.

> **Using automation to monitor the status log:** It is possible to create a form of automation that monitors the status log. The system can send an SNMP trap that can triggers an automation script to issue the **`start`** command to return the state to Consistent Copying or Consistent Synchronized.
>
> Ensure, though, that you understand why the relationship stopped.

Consistent Synchronized

The primary volume is accessible for read and write I/O, and the secondary volume is accessible for read-only I/O. This state is also called “normal copy state” and indicates that your relationship is both consistent and synchronized.

A **stop** command takes the relationship to the Consistent Stopped state. A **stoprcrelationship** command with the **-access** argument takes the relationship to the Idling state.

A **switch** copy direction command leaves the relationship in the Consistent Synchronized state, but reverses the primary and secondary roles and therefore the copy direction.

If the relationship becomes disconnected, the same transitions are made as for the Consistent Stopped state.

Inconsistent Stopped

The primary is accessible for read and write I/O, but the secondary is not accessible for either. The **startrcrelationship** or the **startrcconsistgrp** command must be issued to make the secondary consistent.

This state is entered when the relationship was Inconsistent Copying and has either suffered a persistent error or received a **stop** command. A **start** command causes the relationship to move to the Inconsistent Copying state again.

If the relationship becomes disconnected, the secondary side transitions to Inconsistent Disconnected. The primary side transitions to Idling Disconnected.

Inconsistent Copying

The primary is accessible for read and write I/O, but the secondary is not accessible for either read or write I/O. Inconsistent Copying runs as a background copy process, which copies data from the primary to the secondary.

This state is entered after a **start** command is issued to an Inconsistent Stopped relationship. It is also entered when a forced start is issued to an Idling or Consistent Stopped relationship.

A persistent error or a **stop** command places the relationship into an Inconsistent Stopped state. If the background copy process completes, the relationship transitions to Consistent Synchronized. If the relationship becomes disconnected, the secondary side transitions to Inconsistent Disconnected. The primary side transitions to Idling Disconnected.

Idling

Both master and auxiliary disks operate in the primary role. Both are write accessible and both will journal their writes.

In this state, the relationship accepts a **start** command. The **start** command must specify the new copy direction. A **start** command can cause a loss of consistency if either volume in any relationship has received write I/O, which is indicated by the synchronized status. If the **start** command might lead to a loss of consistency, the **-force** parameter must be specified.

Following a **start** command, the relationship transitions to Consistent Synchronized if no loss of consistency occurs, or to Inconsistent Copying if a loss of consistency does occur.

Also, while in this state, the relationship or the Consistency Group accepts the **-clean** option on the **start** command. If the relationship or the Consistency Group becomes disconnected, both sides change their state to Idling Disconnected.

Idling Disconnected

The volumes in this half of the relationship are in the primary role and accept read and write I/O. The main priority in this state is to recover the link and make the relationship connected again.

Inconsistent Disconnected

The volumes in this half of the relationship are in the secondary role and do not accept read or write I/O.

When the relationship becomes connected again, the relationship will generally become Inconsistent Copying. The relationship will instead transition to Inconsistent Stopped if either the relationship was Inconsistent Stopped when it became disconnected, or the user issued a **stop** command while it was disconnected.

Consistent Disconnected

The volumes in this half of the relationship are all in the secondary role and accept read I/O but not write I/O. This state is entered from Consistent Synchronized or Consistent Stopped when the secondary side of a relationship becomes disconnected.

In this state, the relationship displays an attribute of FreezeTime which is the point in time when consistency was frozen. When entered from Consistent Stopped, it retains the time it had in that state. When entered from Consistent Synchronized, the FreezeTime shows the last time when the relationship was known to be consistent, which corresponds to the time of the last successful heartbeat to the other cluster.

A **stop** command with enable write access to the secondary (**-access** parameter) transitions the relationship to an Idling Disconnected state. This state allows write I/O to be performed to the volumes and is used as part of a disaster recovery scenario.

Empty

This state only applies to a Consistency Group that has no relationships and no other state information to show.

Figure 3-12 shows the various relationship states and events.

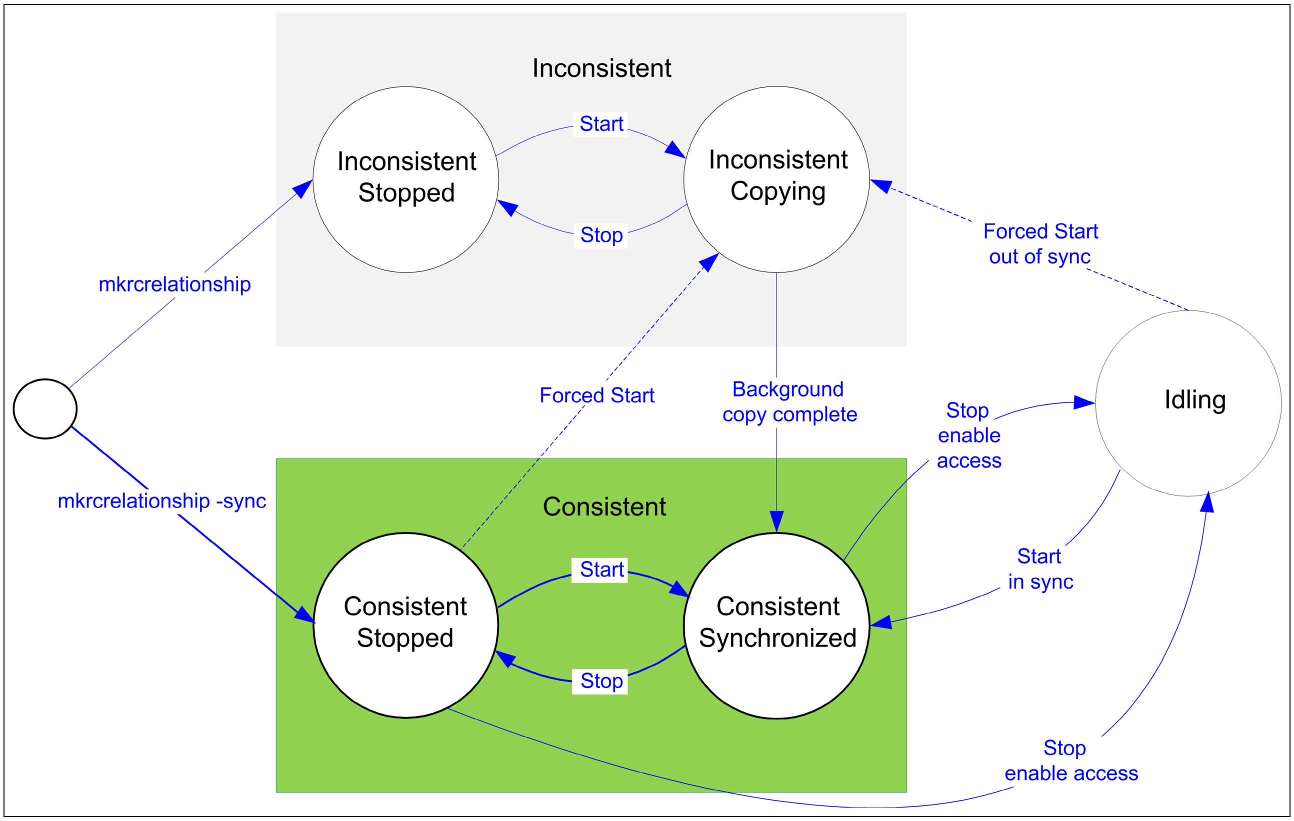


Figure 3-12 Relationship states and events

3.7.3 Stop, Start, and Switch commands

All of these commands are addressed to either a stand-alone relationship or a Consistency Group and are rejected if addressed to a relationship that is part of a Consistency Group. All of these commands accept a parameter of *GroupName* or *RelationshipName,* which is the Consistency Group or relationship to which the command is addressed.

An important aspect of a command's description is whether or not it is *idempotent*. Idempotency means that if an application fails to receive completion for a command, it can retry and reissue the same command without having to check if the command succeeded or failed. If on retry the command returns successfully, the intended end-state has been achieved, although it might have been achieved by the first or second invocation. This situation is described in the following commands.

Stop commands

The **stoprcrelationship** command or the **stoprcconsistgrp** command is issued to stop a relationship, which copies from primary to secondary. If Inconsistent, any copy operation stops and will not resume until a **start** is issued. Write activity will no longer be copied from primary to secondary. For a relationship in the Consistent Synchronized state, this situation causes a consistency freeze.

These commands can be issued to a consistent relationship/Consistency Group (Consistent Stopped, Consistent Synchronized, or Consistent Disconnected) to enable write access to the secondary by selecting the **-access** parameter.

Not selecting the **-access** parameter causes state transitions in the following states:

- Consistent Synchronized becomes Consistent Stopped
- Inconsistent Copying becomes Inconsistent Stopped

Otherwise, the command will be accepted with no state change. When issued to a relationship that is Idling Disconnected, no state change happens, but the relationship does not enter Consistent Synchronized or Inconsistent Copying when the link becomes connected again. Therefore, the command is idempotent.

When the **-access** parameter is selected, the following task is performed:

- Consistent Synchronized or Consistent Stopped becomes Idling
- Consistent Disconnected becomes Idling Disconnected

As a consequence, read/write access to the secondary virtual disks is allowed. If the relationship was Inconsistent, the command fails with no effect (Inconsistent Copying remains Inconsistent Copying).

Start commands

The **startrcrelationship** command or the **startrcconsistgrp** command can only be issued to a relationship that is connected. For a relationship that is in the Idling state, the command assigns a copy direction (primary and secondary roles) and begins the copy process. Otherwise, the command restarts a previous copy process, which was stopped by a **stop** command or an I/O error.

When the resumption of the copy process will lead to a loss of consistency, you must specify the **-force** parameter. When consistency is lost, the secondary cannot be used in the event of a disaster.

In the Idling state, the attribute of the **-primary** parameter (**master** or **aux**) must be provided. In other connected states, the **-primary** parameter can be provided, but it must match the existing setting; otherwise, the command is rejected without effect.

A **start** command requires the **-force** parameter in the following states:

- Consistent Stopped but not synchronized
- Idling but not synchronized

The **-force** parameter is not required, but it can be provided if convenient in one of the following states:

- Inconsistent Stopped
- Inconsistent Copying
- Consistent Synchronized

In the Inconsistent Copying and Consistent Synchronized states, the command has no effect but is accepted to ensure that the start is idempotent.

The **-clean** parameter is used to mark a virtual disk that will become a secondary as clean before issuing the **start** command. This parameter allows you to use an intended secondary that has been initialized by another means, such as from a tape image of the primary. The primary is not declared clean; change recording there allows the secondary to be brought up to date with changes that were made to the primary after the tape image was taken.

This option can only be used as part of a start in the Idling (connected) state. It is accepted without effect if issued in another connected state as part of a valid **start** command.

Switch commands

The **switchrelationship** or **switchrcconsistgrp** command is issued to reverse the roles of primary and secondary in a relationship, perhaps as part of a graceful failover. Write access to the old primary is lost and write access to the new primary is acquired.

These commands can only be issued where the secondary is connected in a consistent state, and where the command does not lead to a loss of consistency.

In other words, the primary and secondary are Synchronized. One of these states must be true:

- Consistent Synchronized
- Consistent Stopped and Synchronized
- Idling and Synchronized

The relationship ends up in the Consistent Synchronized state after this command, with the direction as specified by the **-primary** (as though a **start** command had been issued).

3.8 Tunable parameters and limits

The key tunable parameters are:

- I/O group memory allocation
- Partnership bandwidth
- relationshipbandwidthlimit
- rcbuffersize
- gmlinktolerance and gmmaxhostdelay

3.8.1 I/O Group memory allocation for Metro and Global Mirror

Often referred to as *bitmap* space, the default memory allocation for Metro and Global Mirror on an I/O group is 20 MB. This can be increased up to a maximum of 512 MB, but keep in mind that a maximum of 552 MB applies to the total of all Metro and Global Mirror, and FlashCopy, Volume Mirroring, and RAID management.

A rough estimate is that for each 2 TB of capacity, approximately 1 MB of memory is required for each of the following features:

- RAID
- FlashCopy
- Volume Mirroring
- Metro and Global Mirror

For a detailed breakdown of these numbers, refer to the SAN Volume Controller or Storwize V7000 Infocenters at the following websites and search for **chiogrp**:

http://publib.boulder.ibm.com/infocenter/storwize/ic/index.jsp

http://publib.boulder.ibm.com/infocenter/svc/ic/index.jsp

A default memory value for Metro and Global Mirror of 20 MB will manage around 40 TB of capacity. You can change the I/O group memory allocation.

Example 3-2 shows how to change the remote copy allocation size to 50 MB, which should allow around 100 TB of Metro and Global Mirror.

Example 3-2   Changing the remote copy bitmap size

```
chiogrp -feature remote -size 50 io_group_name
```

If you are using Global Mirror with Change Volumes, also be mindful of the bitmap space available to FlashCopy.

Example 3-3 shows how to change the FlashCopy bitmap allocation size to 50 MB to allow around 100 TB of FlashCopy volumes.

Example 3-3   Changing the FlashCopy bitmap size

```
chiogrp -feature flash -size 50 io_group_name
```

Note that any memory you allocate to these replication features reduces the amount of memory available for general cache.

3.8.2 Partnership bandwidth (total background copy limit)

In 3.4, “Network connectivity for Metro and Global Mirror” on page 42, bandwidth allocation for replication is discussed. The concepts of network bandwidth for normal replication and partnership bandwidth for background copying are related, and generally an allowance of 20 percent is made to allow for background copying for synchronization after an outage.

There are two places where the maximum background copy rate is set. With **mkpartnership**, you can set the upper limit for the *total* of all relationships in a given partnership. With **chsystem**, you can set the upper limit for *each* relationship on a clustered *system* as discussed in 3.8.3, “relationshipbandwidthlimit” on page 58.

When you create a partnership with the **mkpartnership** command, you must specify the unidirectional background copy bandwidth in MBps. There is no default value for this parameter, so you will need to consider the capabilities of your link and your planned workload.

Example 3-4 shows how to set the background copy bandwidth limit to 50 MBps (400 Mbps).

Example 3-4   Setting the background copy bandwidth limit to 50 MBps

```
mkpartnership -bandwidth 50 remote_cluster_name
```

mkpartnership creates a partnership in only one direction, so it needs to be issued from both ends to create a full two-way partnership.

The -**bandwidth** parameter sets the bandwidth that is used by Metro Mirror or Global Mirror for the initial and any subsequent background copy process. If you set this parameter to a value at or above the actual link speed, it will attempt to use the entire link.

After a partnership is established, the **chpartnership** command can be used to change the outgoing one-way total background copy bandwidth limit on a Metro or Global Mirror intercluster partnership.

Example 3-5 shows how to change the total partnership background copy limit to 5 MBps (40 Mbps).

Example 3-5   Setting the total partnership background copy limit to 5 MBps

```
chpartnership -bandwidth 5 remote_cluster_name
```

Remember that to set the background copy bandwidth in the other direction, you will need to issue this command again from the other cluster in the partnership.

The partnership bandwidth allocated will be divided evenly between nodes, and each node will divide its share evenly between relationships, irrespective of capacity or workload differences among relationships. Remember that this only relates to background copying (catch-up/sync). This is not for normal transactional flow traffic.

Bandwidth impact on foreground I/O latency

The background copy bandwidth determines the rate at which the background copy will be attempted. If set too high on a Metro Mirror or Global Mirror relationship, background copy bandwidth can affect server application I/O performance and some relationships might be terminated. This caution does not apply to Global Mirror with Change Volumes.

Note the following points:

- The *network* you provide might impose a practical limit on replication traffic. It is best not to set the background copy rate too high, because it needs to share the available bandwidth with other active relationships. If you want to use a minimum bandwidth network, consider using Global Mirror with Change Volumes.
- The *disk systems* you deploy might impose a practical limit on total I/O. It might be tempting to deploy low performance disk at the disaster recovery (DR) site, but it should at least be able to cope with the peak write traffic coming from the production site. Otherwise, the production server I/O performance might be affected. If you want to use low performing disk at the DR site, consider using Global Mirror with Change Volumes.

> **Partnership bandwidth consideration:** Partnership bandwidth affects how quickly you return to a normal protected state (ConsistentSynchronized) after a network outage. An estimate for this is to be an extra 20 percent on top of the normal transactional flow bandwidth.
>
> However, carefully consider the recovery time and possible limiting factors of any choice you make.

3.8.3 relationshipbandwidthlimit

There are two places where the maximum background copy rate is set, as explained here:

- With **mkpartnership**, you can set the upper limit for the *total* of all relationships in a given *partnership* between two systems as discussed in 3.8.2, “Partnership bandwidth (total background copy limit)” on page 57.
- With **chsystem**, you can set the upper limit for *each* relationship on a clustered *system*.

The relationship bandwidth limit sets the maximum rate for each volume relationship to synchronize. The default setting is 25 MBps, and the valid range is between 1 and 1000 MBps. After it is set, this will be applied to each Metro or Global Mirror relationship.

Example 3-6 shows how you can change the per relationship background copy limit to 3 MBps. This will affect each relationship on the system. This means that no single volume relationship can ever resynchronize at faster than 3 MBps. Remember to issue the command on *both* clusters in the partnership.

Example 3-6 Changing the per relationship limit to 3 MBps

```
chsystem -relationshipbandwidthlimit 3
```

3.8.4 rcbuffersize

rcbuffersize is relevant to Global Mirror. For more information about this topic, refer to 3.10.2, "rcbuffersize" on page 63.

3.8.5 gmlinktolerance and gmmaxhostdelay

gmlinktolerance and **gmmaxhostdelay** are specific to Global Mirror. For more information about this topic, refer to 3.10.1, "gmlinktolerance and gmmaxhostdelay" on page 62.

3.8.6 Configuration limits

Table 3-1 on page 59 shows the Metro Mirror and Global Mirror configuration limits for Version 6.4.

For the latest information, visit the following sites and refer to the Restrictions for your release:

http://www-03.ibm.com/systems/storage/software/virtualization/svc/interop.html

http://www-03.ibm.com/systems/storage/disk/storwize_v7000/interop.html

Table 3-1 Metro Mirror and Global Mirror configuration limits

Parameter	Value
Metro or Global Mirror Consistency Groups	256 per system.
Metro Mirror and Global Mirror relationships	2048 per I/O Group.
Number of Global Mirror with Change Volumes relationships	256 per system.
Total capacity of all Metro and Global Mirror master, auxiliary, and change volumes	1024 TB per I/O group. See 3.8.1, "I/O Group memory allocation for Metro and Global Mirror" on page 56.
Maximum size of a volume for Global Mirror or Metro Mirror	256 TB.
Maximum size of a volume for Global Mirror with Change Volumes	Currently 2 TB (as at 6.4.0), but this might be increased in future releases. Check the latest documentation for current information.
Metro Mirror and Global Mirror supported with VMware Site Recovery Manager?	Yes. For details, see: http://www.vmware.com/resources/compatibility/search.php?action=base&deviceCategory=san
Total number of inter-cluster partnerships in a connected set	4; that is, a system may be connected to three others.

Configuration limit consideration: Some configuration limits apply per cluster (system). Some configuration limits apply per I/O group.

3.9 Metro Mirror

Metro Mirror provides synchronous replication. It is designed to ensure that updates are committed to both the primary and secondary volumes before sending an acknowledgement (Ack) of the completion to the server.

If the primary volume fails completely for any reason, Metro Mirror is designed to ensure that the secondary volume will hold the same data as the primary did immediately prior to the failure.

Metro Mirror provides the simplest way to maintain an identical copy on both the primary and secondary volumes. However, as with any synchronous copy over long distance, there can be a performance impact to host applications due to network latency.

Metro Mirror supports relationships between volumes that are up to 300 km apart. Latency is an important consideration for any Metro Mirror network. With typical fiber optic round-trip latencies of 1 ms per 100 km you can expect a minimum of 3 ms additional latency, due to the network alone, on each I/O if you are running across 300 km separation.

Figure 3-13 on page 60 shows the order of Metro Mirror write operations.

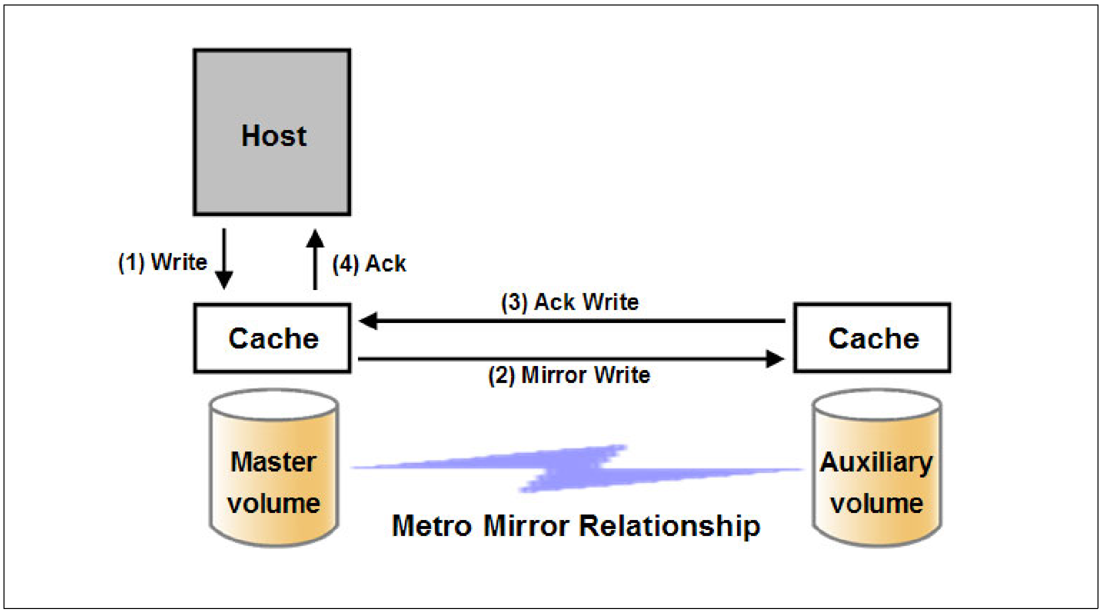


Figure 3-13 Metro Mirror write sequence

Note that a write into mirrored cache on an SVC or Storwize V7000 system is all that is required for the write to be considered as committed. Destaging to disk is a natural part of I/O management, but it is not generally in the critical path for a Metro Mirror write acknowledgement.

ISL hop count: When a local fabric and a remote fabric are connected together for Metro Mirror purposes, the inter-switch link (ISL) hop count between a local node and a remote node should not exceed seven hops.

3.10 Global Mirror

Global Mirror provides asynchronous replication. It is designed to reduce the dependency on round-trip network latency by acknowledging the primary write in parallel with sending the write to the secondary volume.

If the primary volume fails completely for any reason, Global Mirror is designed to ensure that the secondary volume will hold the same data as the primary did at a point a short time prior to the failure. That short period of data loss might typically be between 10 milliseconds and 10 seconds, but will vary according to individual circumstances.

Global Mirror provides a way to maintain a write-order-consistent copy of data at a secondary site only slightly behind the primary and with minimal impact on the performance of the primary volume.

Global Mirror supports relationships between volumes with up to 80 ms round-trip latency. Based on the 1 ms per 100 km estimate, this suggests that the two sites could be separated by up to 8600 km using a high quality network. However, many commercial networks have peak latencies in the tens of milliseconds over relatively short distances.

When a write is sent to a primary volume, it is also assigned a sequence number so that writes sent to the secondary are committed in the original order. This is a way of guaranteeing write order consistency even when multiple writes are issued for the same block.

A common issue among asynchronous replication technologies is the question of overlapping writes, sometimes called *colliding writes*. This is something all asynchronous replication systems need to deal with, in one way or another. An overlapping write is when a write is received from a host while a previous secondary write is still active for the same block.

In previous versions of SVC, writes were serialized by a sector lock, so SVC would cause the new write to wait until the previous one had been completed. SVC and Storwize V7000 now assign sequence numbers for each write so multiple outstanding writes are now allowed for the same disk sector. Managing write sequencing does, however, impose additional latency on the write I/Os.

Figure 3-14 shows the order of Global Mirror write operations.

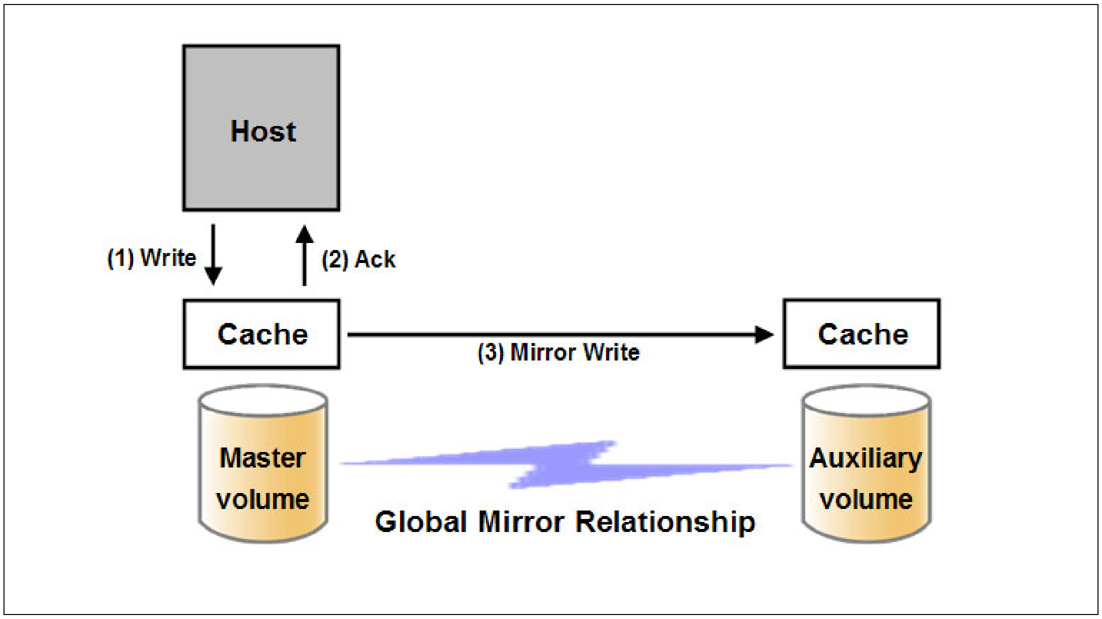


Figure 3-14 Global Mirror write sequence

Note that a write into mirrored cache on an SVC or Storwize V7000 system is all that is required for the write to be considered as committed. Destaging to disk is a natural part of I/O management, but it is not generally in the critical path for a write acknowledgement.

3.10.1 gmlinktolerance and gmmaxhostdelay

The **gmlinktolerance** and **gmmaxhostdelay** parameters are critical in the system for deciding internally whether to terminate a relationship due to a performance problem. In most cases these two parameters need to be considered in tandem, and the defaults would not normally be changed unless you had a specific reason to do so.

- The default setting for **gmlinktolerance** is 300 seconds (5 minutes).
- The default setting for **gmmaxhostdelay** is 5 milliseconds.

gmmaxhostdelay can be thought of as the maximum host I/O impact that is due to Global Mirror, that is, how long would that local I/O take with Global Mirror turned off, and how long does it take with Global Mirror turned on. The difference is the host delay due to Global Mirror tag and forward processing.

gmlinktolerance can be thought of as how long you allow the host delay to go on being significant before you decide to terminate a Global Mirror volume relationship.

Increasing either or both of these parameters could impact host performance. Decreasing either or both could improve host performance but lead to relationship terminations (1920 errors).

Parameters like these can help to ensure your system runs smoothly. However, they cannot compensate for insufficient bandwidth or excessive network latency, or if all of your quality of service channels have been allocated elsewhere.

Although the default settings are adequate for most situations, increasing one parameter while reducing another might deliver a tuned performance environment for a particular circumstance.

For example, Global Mirror is designed to look at average delays. However, some hosts such as VMware ESX might not tolerate a single I/O getting old, for example, 45 seconds, before it decides to reboot. Given that it is better to terminate a Global Mirror relationship than it is to reboot a host, you might want to set **gmlinktolerance** to something like 30 seconds and then compensate, so that you do not get too many relationship terminations by setting **gmmaxhostdelay** to something larger such as 100 ms.

If you compare the two approaches, the default (**gmlinktolerance 300**, **gmmaxhostdelay 5**) is saying "If more than half the I/Os are slow and that happens repeatedly for 5 minutes, then terminate the busiest relationship in that stream." In contrast, the example of **gmlinktolerance 30**, **gmmaxhostdelay 100** says "If more than half the I/Os are extremely slow and that happens repeatedly for 30 seconds, then terminate the busiest relationship in the stream."

So one approach is designed to pick up general slowness, and the other approach is designed to pick up shorter bursts of extreme slowness that could disrupt your server environment.

As shown in Example 3-7, if your hosts are particularly sensitive to an occasional long-outstanding write, you could set **gmlinktolerance 30** and **gmmaxhostdelay 100**, but it is strongly advisable to thoroughly test any such changes in a non-production environment.

Example 3-7 Changing gmlinktolerance to 30 and gmmaxhostdelay to 100

```
chsystem -gmlinktolerance 30
chsystem -gmmaxhostdelay 100
```

Test and monitor: To reiterate, thoroughly test and carefully monitor the host impact of any changes like this before putting them into a live production environment.

gmlinktolerance can be set between 20 seconds and 86400 seconds, or it can be set to zero to *disable* it. Disabling **gmlinktolerance** is generally unwise because it will cause your Global Mirror relationships to stay up no matter what impact that might have on host applications.

The following explanation approximates how the **gmlinktolerance** and **gmmaxhostdelay** are used together:

1. Check every 10 seconds to see whether more than a third of the I/Os in any one stream exceeded the **gmmaxhostdelay**.
2. If more than a third of the I/Os were slow, then increase a counter by one for that stream. Otherwise, decrease the counter by one.
3. If the counter gets to **gmlinktolerance**/10, terminate the busiest relationship in the stream and issue event code 1920.

3.10.2 rcbuffersize

rcbuffersize was introduced with the Version 6.2 code level so that systems with extremely intense and bursty write I/O would not fill the internal buffer while Global Mirror writes were undergoing sequence tagging.

Important: Do not change the **rcbuffersize** parameter except under the direction of IBM Support.

Example 3-8 shows how to change **rcbuffersize** to 64 MB using the **chsystem** command. The default value for **rcbuffersize** is 48 MB and the maximum is 512 MB.

Example 3-8 Changing rcbuffersize to 64 MB

```
chsystem -rcbuffersize 64
```

Remember that any additional buffers you allocate will be taken away from general cache.

3.10.3 Link delay simulation

Even though Global Mirror is an asynchronous replication method, there can be an impact to server applications due to Global Mirror managing transactions and maintaining write order consistency over a network.

To mitigate this, as a testing and planning feature, Global Mirror allows you to simulate the effect of the round-trip delay between sites. This can be enabled for intra-cluster or inter-cluster Global Mirror to allow you to predict possible performance impacts on your server applications when you deploy a long distance Global Mirror.

The default for this parameter is zero and the valid range is from zero to 100 milliseconds of round-trip delay. Remember that the maximum round-trip delay supported on a live Global Mirror is 80 milliseconds.

Example 3-9 shows how to change both the inter-system and the intra-system delays to 80 milliseconds.

Example 3-9 Changing the delay simulation parameters to 80 milliseconds

```
chcluster -gminterdelaysimulation 80
chcluster -gmintradelaysimulation 80
```

3.11 Global Mirror with Change Volumes

Global Mirror with Change Volumes provides asynchronous replication based on point-in-time copies of data. It is designed to allow for effective replication over lower bandwidth networks and to reduce any impact on production hosts.

Metro Mirror and Global Mirror both require the bandwidth to be sized to meet the peak workload, whereas Global Mirror with Change Volumes requires only to be sized to meet the average workload across a cycle period.

Figure 3-15 shows a high level conceptual view of Global Mirror with Change Volumes. GM/CV uses FlashCopy to maintain image consistency and to isolate host volumes from the replication process.

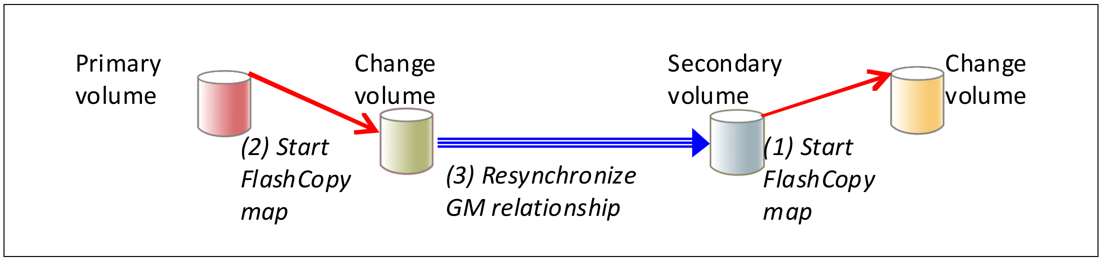


Figure 3-15 Global Mirror with Change Volumes

Global Mirror with Change Volumes will also only send one copy of a changed grain which might have been rewritten many times within the given cycle period.

If the primary volume fails completely for any reason, GM/CV is designed to ensure that the secondary volume will hold the same data as the primary did at a given point in time. That period of data loss might typically be between 5 minutes and 24 hours, but will vary according to the design choices you make.

Change Volumes hold point-in-time copies of 256 KB grains. If there is a change to any of the disk blocks in a given grain, that grain will be copied to the change volume to preserve its contents. Change Volumes are also maintained at the secondary site so that a consistent copy of the volume is always available even when the secondary volume is being updated.

Primary and Change Volumes are always in the same I/O group and the Change Volumes are always thin-provisioned. Change Volumes cannot be mapped to hosts and used for host I/O, and they cannot be used as a source for any other FlashCopy or Global Mirror operations.

Figure 3-16 on page 66 shows how a Change Volume is used to preserve a point-in-time data set, which is then replicated to a secondary site. The data at the secondary site is in turn preserved by a Change Volume until the next replication cycle has completed.

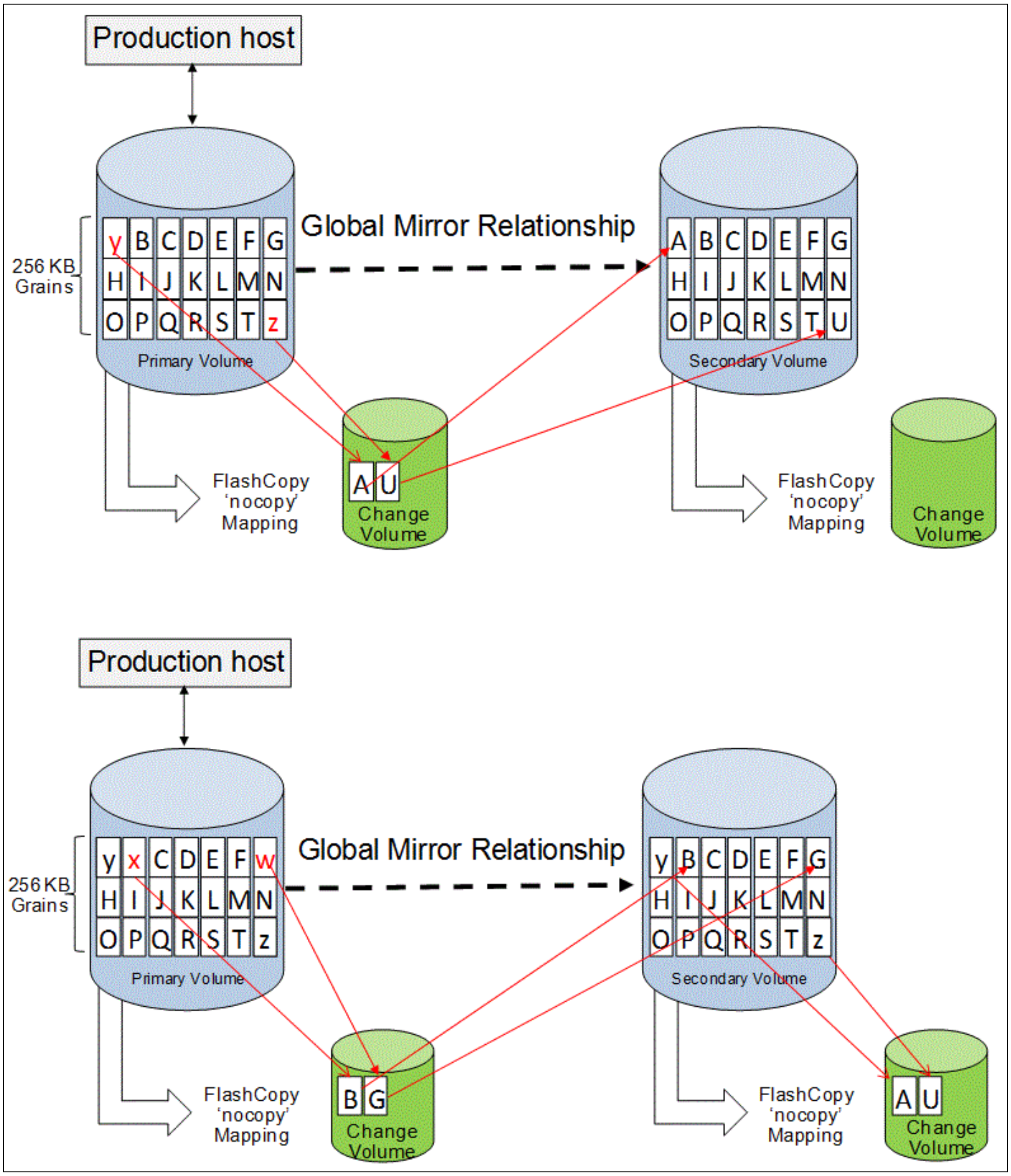


Figure 3-16 Global Mirror with Change Volumes uses FlashCopy point-in-time copy technology

> **FlashCopy mapping note:** These FlashCopy mappings are not standard FlashCopy volumes and are not accessible for general use. They are internal structures dedicated to supporting Global Mirror with Change Volumes.

The options for **`-cyclingmode`** are none | multi.

Specifying or taking the default **`none`** will mean that Global Mirror acts in its traditional mode without Change Volumes.

Specifying **multi** will mean that Global Mirror will start cycling based on the cycle period, which defaults to 300 seconds. The valid range is from 60 seconds to 24*60*60 seconds (86 400 seconds = one day). If all of the changed grains cannot be copied to the secondary site within the specified time, then the replication is designed to take as long as it needs and to start the next replication as soon as the earlier one has finished. You can choose to implement this approach by deliberately setting the cycle period to a short amount of time, which is a perfectly valid approach. Remember, however, that the shorter the cycle period, the less opportunity there is for peak write I/O smoothing, and the more bandwidth you will need. Figure 3-17 illustrates setting the cycle period.

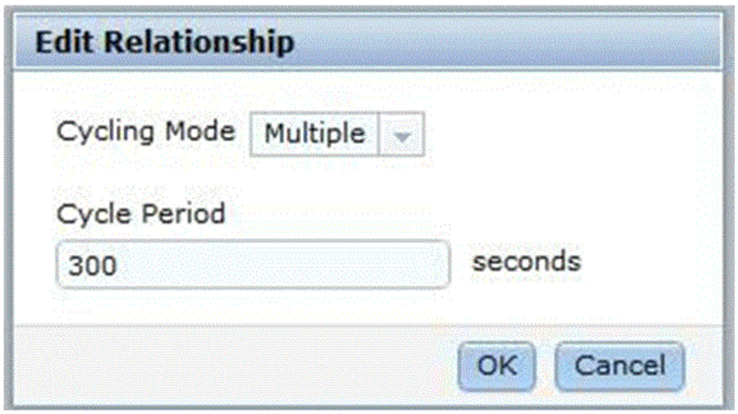


Figure 3-17 Setting the cycle period

Note that the **cyclingmode** can only be changed when the Global Mirror relationship is in a stopped state.

3.11.1 Recovery point objective using Change Volumes

Recovery point objective (RPO) is the maximum tolerable period in which data might be lost if you switch over to your secondary volume.

If a cycle completes within the specified cycle period, then the RPO will be not more than 2 x cycle period. However, if it does not complete within the cycle period, then the RPO is not more than the sum of the last two cycle times.

The current RPO can be determined by looking at the **lsrcrelationship** freeze time attribute. The freeze time is the time stamp of the last primary Change Volume that has completed copying to the secondary site. Note the following example:

1. The cycle period is the default of 5 minutes and a cycle is triggered at 6:00 a.m.
2. At 6:03 a.m., the cycle completes.
3. The freeze time would be 6:00 a.m. and the RPO at this point would be 3 minutes.
4. The cycle starts again at 6:05 a.m. and the RPO at this point would be 5 minutes.
5. The cycle is still running at 6:12 a.m. and the RPO is now up to 12 minutes, because 6:00 a.m. is still the freeze time of the last complete cycle.
6. At 6:13 a.m., the cycle completes and the RPO at this point would be 8 minutes because 6:05 a.m. is the freeze time of the last complete cycle.
7. Because the cycle period has been exceeded, the cycle immediately starts again.

3.11.2 Bandwidth sizing example for Change Volumes

As an example, consider a business with the following I/O profile:

- Average write size 8 kilobytes (= 8 x 8 bits/1024 = 0.0625 Mbits).
- For most of the day between 8 a.m. and 8 p.m. the write activity is around 1500 writes per second.
- Twice a day, once in the morning and once in the afternoon, the system bursts up to 4500 writes per second for up to 10 minutes.
- Outside of the 8 a.m. to 8 p.m. window there is little or no I/O write activity.

This example is intended to represent a general traffic pattern that might be common in many medium-sized sites.

Here we consider options for Metro Mirror, Global Mirror, and for Global Mirror with Change Volumes based on a cycle period of 30 minutes and 60 minutes.

Metro Mirror or Global Mirror would require bandwidth on the instantaneous peak of 4500 writes per second as follows:

```
4500 x 0.0625 = 282 Mbps + 20% resync allowance + 3 Mbps heartbeat
= 341 Mbps dedicated plus any safety margin plus growth
```

In the following two examples, the bandwidth for GM/CV needs to be able to handle the peak 30-minute period, or the peak 60-minute period.

Peak 30-minute period example

If we look at this broken into 10-minute periods, the peak 30-minute period is made up of one 10-minute period of 4500 writes per second, and two 10-minute periods of 1500 writes per second. The average write rate for the 30-minute cycle period can then be expressed mathematically as:

```
(4500 + 1500 + 1500) / 3 = 2500 writes/sec for a 30-minute cycle period
```

The minimum bandwidth required for the cycle period of 30 minutes is:

```
2500 x 0.0625 = 157 Mbps + 20% resync allowance + 3 Mbps heartbeat
= 192 Mbps dedicated plus any safety margin plus growth
```

Peak 60-minute period example

For a cycle period of 60 minutes, the peak 60-minute period is made up of one 10-minute period of 4500 writes per second, and five 10-minute periods of 1500 writes per second. The average write for the 60-minute cycle period can be expressed as:

```
(4500 + 5 x 1500) / 6 = 2000 writes/sec for a 60-minute cycle period
```

The minimum bandwidth required for a cycle period of 60 minutes is:

```
2000 x 0.0625 = 125 Mbps + 20% resync allowance + 3 Mbps heartbeat
= 153 Mbps dedicated plus any safety margin plus growth
```

Now consider if the business does not have aggressive RPO requirements and does not want to provide dedicated bandwidth for Global Mirror, but the network is available and unused at night so Global Mirror can use that. There is an element of risk here, which is if the network is unavailable for any reason, GM/CV cannot keep running during the day until it catches up, so you would need to allow a much higher resync allowance in your replication window, for example, 100 percent. A GM/CV replication based on daily point-in-time copies at 8 p.m. each night and replicating until 8 a.m. at the latest would probably require bandwidth of at least:

```
(9000 + 70 x 1500) / 72 = 1584 x 0.0625 = 99 Mbps + 100% + 3 Mbps heartbeat
= 201 Mbps at night plus any safety margin plus growth, non-dedicated,
time-shared with daytime traffic
```

Global Mirror with Change Volumes provides a way to maintain point-in-time copies of data at a secondary site where there is insufficient bandwidth to replicate the peak workloads in real-time.

Another factor that can reduce the bandwidth required for Global Mirror with Change Volumes is that it will also only send one copy of a changed grain, which might have been rewritten many times within the given cycle period.

Remember that these are simply examples. The central principle of sizing is that you need to know your data write rate, that is, the number of write I/Os and the average size of those I/Os. For Metro Mirror and Global Mirror you need to know the peak write I/O rates. For GM/CV, you need to know the average write I/O rates.

3.11.3 Disk sizing for Change Volumes

The Change Volumes must be created to be the same size as the primary volumes, but because they are thin-provisioned, the actual used space can remain quite small.

There is no definitive sizing rule, because each customer will have different data access patterns, but for typical cycle times of an hour or less, disk sizing is likely to be only a minor consideration. For example, if 10 percent of all grains were changed in a day, then Change Volumes cycling every hour would use perhaps only a small percent of additional disk space.

Disk sizing will also depend on the working set size of the data. That is, if the same grains are rewritten over and over during the day, then 10 percent changed grains in a day could also mean 10 percent changed data in an hour. If the Change Volumes are set to cycle only one time in 24 hours, then the Change Volumes are much more likely to use 10 percent additional disk space.

A more accurate way to estimate disk sizing is to base it on the write rate at the peak cycle, for example, the peak hour if you are using a one-hour cycle period, and multiply it by the average I/O size. So the example in 3.11.2, “Bandwidth sizing example for Change Volumes” on page 68 gives the following figures:

```
2000 writes per second x 8 kilobytes x 3600 seconds
      = 57 600 000 bytes per hour
      = 57.6 Megabytes per hour
```

So in this example, you would need only a small amount of actual disk space for the Change Volume.

You will soon learn how much disk space is required in your particular environment. And although it is likely to be quite a small percentage, it is unwise to start a Global Mirror with Change Volumes implementation if you do not have any spare space.

3.11.4 Migrating from Global Mirror to Global Mirror with Change Volumes

Global Mirror relationships can be easily changed to Global Mirror with Change Volumes by using the GUI. Right-click the relationship you want to change and select **Create New**.

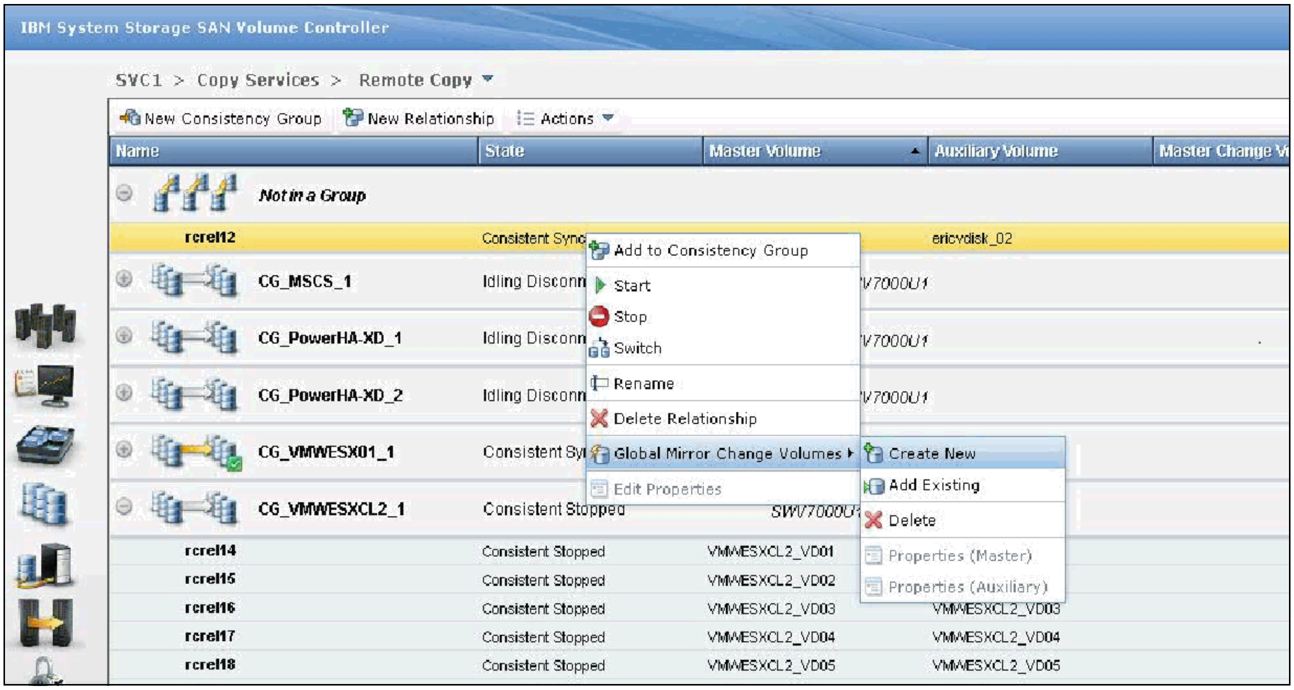


Figure 3-18 Select Create New

The GUI will prompt you to add both Master and Auxiliary volumes (for example, local and remote volumes); see Figure 3-19.

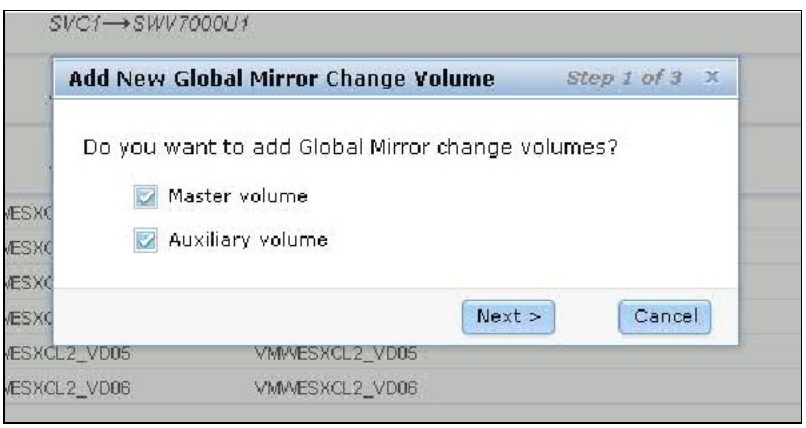


Figure 3-19 Add Master volume and Auxiliary volume

Choose to use an existing volume for both, and the system will automatically create the change volumes behind the scenes; see Figure 3-20 on page 71.

Figure 3-20 Use an existing volume for the Master volume

Figure 3-21 illustrates selecting an existing volume for the Auxiliary volume.

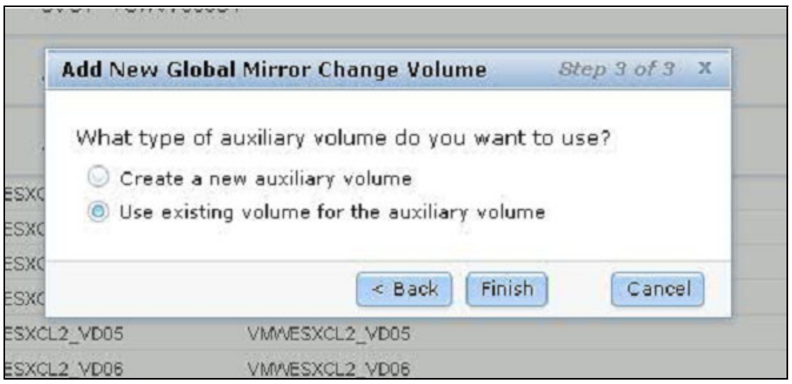


Figure 3-21 Use an existing volume for the Auxiliary volume

3.12 Intra-cluster replication

Replication can be an *intra-cluster* relationship between two nodes within the same I/O group, or an *inter-cluster* relationship between two separate systems.

Intra-cluster replication is likely to be used for testing or training purposes because it requires less equipment. An additional use case for intra-cluster Metro Mirror might be for converting volumes from one extent size to a larger extent size, but this would generally be done using Volume Mirroring.

The background copy rate on intra-cluster replication is fixed at 25 MBps for each node. For more information, see also 3.10.3, "Link delay simulation" on page 64.

System overhead consideration: Be aware that intra-cluster Global Mirror has higher system overheads than intra-cluster Metro Mirror. Thus, the only use case for intra-cluster Global Mirror would be for testing or training.

3.13 Three-site replication with SVC Stretched Cluster and Metro Mirror or Global Mirror

SVC Stretched Cluster allows for two nodes within a single SAN Volume Controller I/O group to be separated by up to 300 kilometres in some configurations. Keep in mind that 100 Km separation will add up to 2 ms in round-trip read transaction latency (two round-trip I/Os) and 5 ms in round-trip write transaction latency (five round-trip I/Os).

Table 3-2 shows supported FC distances for SVC Stretched Cluster without private ISLs between nodes.

Table 3-2 Supported FC distances for SVC Stretched Cluster without private ISLs between nodes

Maximum distance	Maximum link speed
10 Km	8 Gbps
20 Km	4 Gbps
40 Km	2 Gbps

Table 3-3 shows supported FC distances for SVC Stretched Cluster with private ISLs between nodes.

Table 3-3 Supported FC distances for SVC Stretched Cluster with private ISLs between nodes

Maximum distance	Usage
100 Km	Failover, Vmotion, Live Partition Mobility
300 Km	Failover

Stretched Cluster allows the SVC to mitigate against the loss of a site and can also be used to enhance virtualized environments such as VMware VMotion and AIX Live Partition Mobility and failover solutions such as Microsoft Clusters. Failover does not require scripting or zoning changes, or manual intervention.

A key requirement is always to place the primary quorum on a third-power domain (or preferably, at a third site). SVC will also now support an option to place the primary quorum at the far end of a much longer link, using FCIP, with a maximum round-trip latency of 80 milliseconds.

From a Metro Mirror and Global Mirror point of view, the third site, which traditionally only houses the primary quorum, could also be a Metro Mirror or Global Mirror site, thereby providing a three-site replication solution.

Figure 3-22 on page 73 shows a traditional SVC Stretched Cluster.

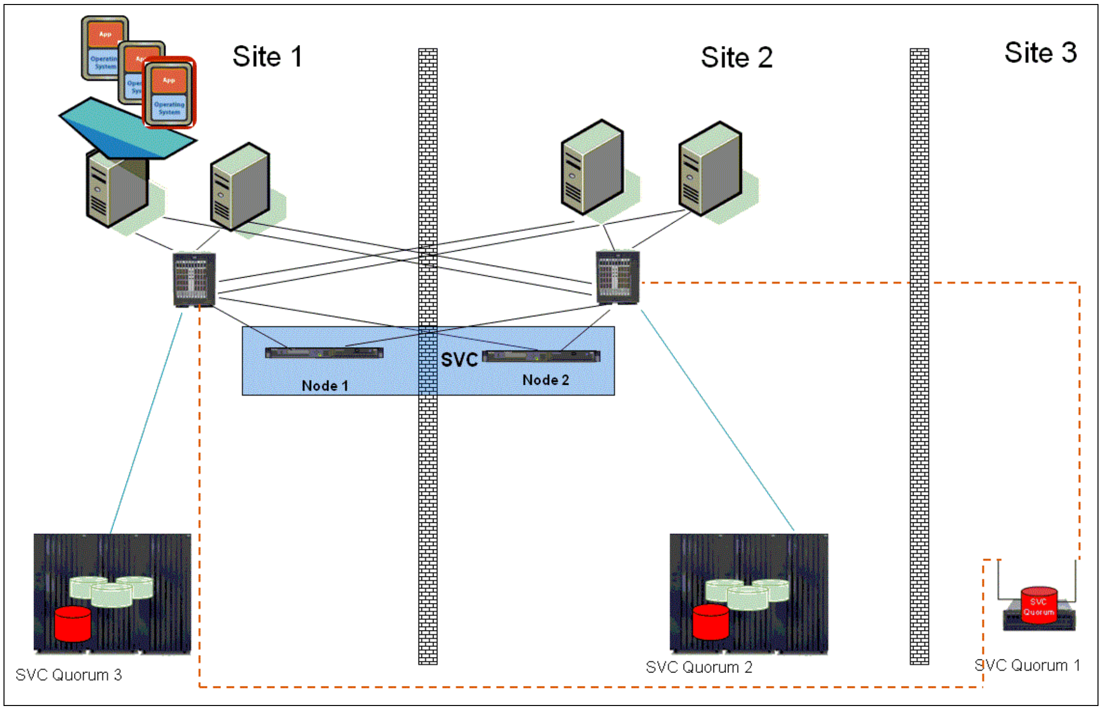


Figure 3-22 Traditional Stretched Cluster high availability solution

Figure 3-23 shows how a Stretched Cluster high availability solution can be augmented with Metro Mirror or Global Mirror to produce a three-site replication solution.

Stretched Cluster will provide automated failover to a secondary site. Global Mirror failover will require manual intervention.

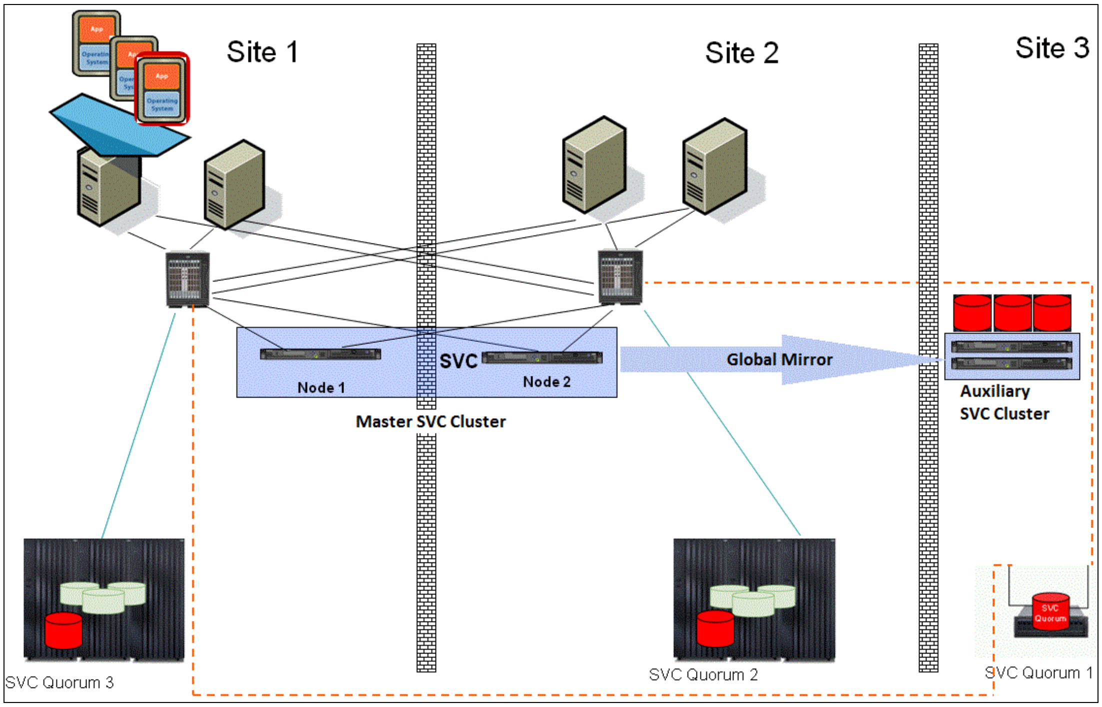


Figure 3-23 Using Stretched Cluster with Metro or Global Mirror to create a three-site replication solution

Both of the Stretched Cluster I/O group sites must be connected to the Metro Mirror or Global Mirror site.

If private ISLs between nodes are used, Metro Mirror or Global Mirror is only attached to the public FC network.

For more details about Stretched Cluster see Chapter 7, “Volume Mirroring” on page 275 and Chapter 12, “Fabric design considerations for Replication Family” on page 473.

SVC Stretched Cluster I/O Group: Although SVC Stretched Cluster is the preferred naming convention, it is more completely described as SVC Split I/O Group. The I/O Groups must be split to benefit from Volume Mirroring automated failover.

4

Implementing Metro Mirror and Global Mirror

Through the use of examples, in this chapter we explain and illustrate how to implement Metro Mirror and Global Mirror by using the command-line interface (CLI) and by using the graphical user interface (GUI).

The following topics are addressed:

- Metro Mirror using the CLI
- Global Mirror using the CLI
- Global Mirror with Change Volumes using the CLI
- Implementation using the GUI

4.1 Metro Mirror using the CLI

> **Cluster consideration:** This example is for inter-cluster Metro Mirror.
>
> If you want to set up *intra-cluster* operations, we highlight those parts of the following procedure that you do *not* need to perform.

In the following scenario, we set up an inter-cluster Metro Mirror relationship between the SVC system `ITSO_SVC1` primary site and the SVC system `ITSO_SVC4` at the secondary site. Table 4-1 shows the details of the volumes.

Table 4-1 Volume details

Content of volume	Volumes at primary site	Volumes at secondary site
Database files	MM_DB_Pri	MM_DB_Sec
Database log files	MM_DBLog_Pri	MM_DBLog_Sec
Application files	MM_App_Pri	MM_App_Sec

Because data consistency is needed across the `MM_DB_Pri` and `MM_DBLog_Pri` volumes, a `CG_WIN2K3_MM` Consistency Group is created to handle Metro Mirror relationships for them.

Because in this scenario the application files are independent of the database, a stand-alone Metro Mirror relationship is created for the `MM_App_Pri` volume. Figure 4-1 illustrates the Metro Mirror setup.

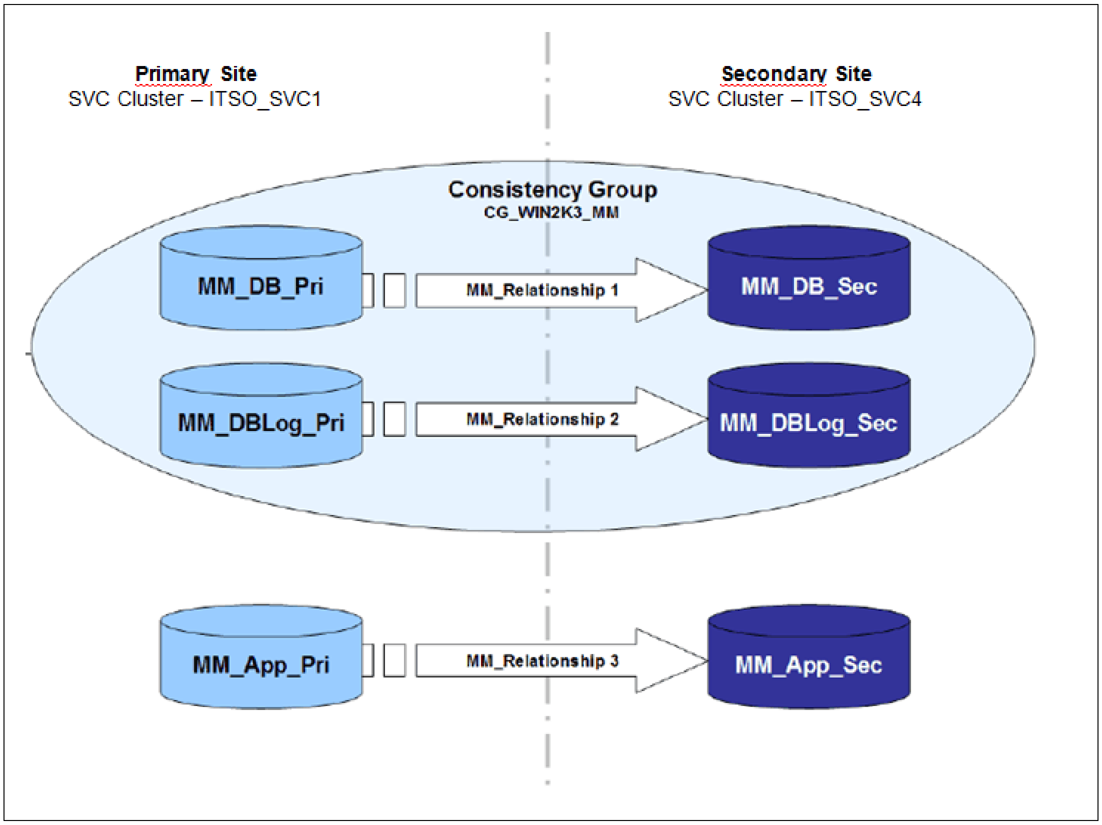


Figure 4-1 Metro Mirror scenario

4.1.1 Setting up Metro Mirror

In the following section, we assume that the source and target volumes have already been created and that the inter-switch links (ISLs) and zoning are in place, enabling the SVC clustered systems to communicate.

To set up the Metro Mirror, perform the following steps:

1. Create an SVC partnership between `ITSO_SVC1` and `ITSO_SVC4` on both of the SVC clustered systems.
2. Create a Metro Mirror Consistency Group:

   Name: `CG_W2K3_MM`
3. Create the Metro Mirror relationship for `MM_DB_Pri`:
   – Master: `MM_DB_Pri`
   – Auxiliary: `MM_DB_Sec`
   – Auxiliary SVC system: `ITSO_SVC4`
   – Name: `MMREL1`
   – Consistency Group: `CG_W2K3_MM`
4. Create the Metro Mirror relationship for `MM_DBLog_Pri`:
   – Master: `MM_DBLog_Pri`
   – Auxiliary: `MM_DBLog_Sec`
   – Auxiliary SVC system: `ITSO_SVC4`
   – Name: `MMREL2`
   – Consistency Group: `CG_W2K3_MM`
5. Create the Metro Mirror relationship for `MM_App_Pri`:
   – Master: `MM_App_Pri`
   – Auxiliary: `MM_App_Sec`
   – Auxiliary SVC system: `ITSO_SVC4`
   – Name: `MMREL3`

In the following section, we perform each step by using the CLI.

4.1.2 Creating a partnership - Metro Mirror

We create the partnership on both systems.

> **Intra-cluster consideration:** If you are creating an intra-cluster Metro Mirror, do *not* perform the next step; instead, go to “Creating a Metro Mirror Consistency Group” on page 81.

Verifying that the planned partner is visible in the network

To verify that both systems can communicate with each other, use the **`lspartnershipcandidate`** command.

As shown in Example 4-1, `ITSO_SVC4` is an eligible SVC system candidate at `ITSO_SVC1` for the system partnership, and vice versa. Therefore, both systems communicate with each other.

Example 4-1 Listing the systems available for partnership

```
IBM_2145:ITSO_SVC1:admin>lspartnershipcandidate
id               configured name
0000020061C06FCA no         ITSO_SVC4
```

```
000002006AC03A42 no         ITSO_SVC2
0000020060A06FB8 no         ITSO_SVC3
00000200A0C006B2 no         ITSO-Storwize-V7000-2

IBM_2145:ITSO_SVC4:admin>lspartnershipcandidate
id               configured name
000002006AC03A42 no         ITSO_SVC2
0000020060A06FB8 no         ITSO_SVC3
00000200A0C006B2 no         ITSO-Storwize-V7000-2
000002006BE04FC4 no         ITSO_SVC1
```

Example 4-2 shows the output of the **lspartnership** and **lssystem** commands, before setting up the Metro Mirror relationship. We show them so that you can compare with the same relationship after setting up the Metro Mirror relationship.

As of code level 6.3, you can create a partnership between an SVC system and a Storwize V7000 system if you first change the *layer* setting on the Storwize V7000 from **storage** to **replication** with the **chsystem -layer** command. This option can only be used if no other systems are visible on the fabric, and no system partnerships are defined. SVC systems are always in the *appliance* layer, as shown in Example 4-2.

Example 4-2 Pre-verification of system configuration

```
IBM_2145:ITSO_SVC1:admin>lspartnership
id               name      location partnership bandwidth
000002006BE04FC4 ITSO_SVC1 local

IBM_2145:ITSO_SVC4:admin>lspartnership
id               name      location partnership bandwidth
0000020061C06FCA ITSO_SVC4 local

IBM_2145:ITSO_SVC1:admin>lssystem
id 000002006BE04FC4
name ITSO_SVC1
location local
partnership
bandwidth
total_mdisk_capacity 766.5GB
space_in_mdisk_grps 766.5GB
space_allocated_to_vdisks 0.00MB
total_free_space 766.5GB
total_vdiskcopy_capacity 0.00MB
total_used_capacity 0.00MB
total_overallocation 0
total_vdisk_capacity 0.00MB
total_allocated_extent_capacity 1.50GB
statistics_status on
statistics_frequency 15
cluster_locale en_US
time_zone 520 US/Pacific
code_level 6.3.0.0 (build 54.0.1109090000)
console_IP 10.18.228.81:443
id_alias 000002006BE04FC4
gm_link_tolerance 300
gm_inter_cluster_delay_simulation 0
gm_intra_cluster_delay_simulation 0
gm_max_host_delay 5
email_reply
email_contact
```

```
email_contact_primary
email_contact_alternate
email_contact_location
email_contact2
email_contact2_primary
email_contact2_alternate
email_state stopped
inventory_mail_interval 0
cluster_ntp_IP_address
cluster_isns_IP_address
iscsi_auth_method chap
iscsi_chap_secret passw0rd
auth_service_configured no
auth_service_enabled no
auth_service_url
auth_service_user_name
auth_service_pwd_set no
auth_service_cert_set no
auth_service_type tip
relationship_bandwidth_limit 25
tier generic_ssd
tier_capacity 0.00MB
tier_free_capacity 0.00MB
tier generic_hdd
tier_capacity 766.50GB
tier_free_capacity 766.50GB
has_nas_key no
layer appliance

IBM_2145:ITSO_SVC4:admin>lssystem
id 0000020061C06FCA
name ITSO_SVC4
location local
partnership
bandwidth
total_mdisk_capacity 768.0GB
space_in_mdisk_grps 0
space_allocated_to_vdisks 0.00MB
total_free_space 768.0GB
total_vdiskcopy_capacity 0.00MB
total_used_capacity 0.00MB
total_overallocation 0
total_vdisk_capacity 0.00MB
total_allocated_extent_capacity 0.00MB
statistics_status on
statistics_frequency 15
cluster_locale en_US
time_zone 520 US/Pacific
code_level 6.3.0.0 (build 54.0.1109090000)
console_IP 10.18.228.84:443
id_alias 0000020061C06FCA
gm_link_tolerance 300
gm_inter_cluster_delay_simulation 0
gm_intra_cluster_delay_simulation 0
gm_max_host_delay 5
email_reply
email_contact
email_contact_primary
email_contact_alternate
email_contact_location
```

```
email_contact2
email_contact2_primary
email_contact2_alternate
email_state stopped
inventory_mail_interval 0
cluster_ntp_IP_address
cluster_isns_IP_address
iscsi_auth_method none
iscsi_chap_secret
auth_service_configured no
auth_service_enabled no
auth_service_url
auth_service_user_name
auth_service_pwd_set no
auth_service_cert_set no
auth_service_type tip
relationship_bandwidth_limit 25
tier generic_ssd
tier_capacity 0.00MB
tier_free_capacity 0.00MB
tier generic_hdd
tier_capacity 0.00MB
tier_free_capacity 0.00MB
has_nas_key no
layer appliance
```

Establishing the partnership

In Example 4-3, a partnership is created between `ITSO_SVC1` and `ITSO_SVC4`, specifying 50 MBps bandwidth to be used for the background copy.

To check the status of the newly created partnership, issue the **`lspartnership`** command. Also notice that the new partnership is only *partially* configured. It remains partially configured until the Metro Mirror partnership is created on the other node.

Example 4-3 Creating the partnership from ITSO_SVC1 to ITSO_SVC4 and verifying it

```
IBM_2145:ITSO_SVC1:admin>mkpartnership -bandwidth 50 ITSO_SVC4
IBM_2145:ITSO_SVC1:admin>lspartnership
id               name      location partnership                 bandwidth
000002006BE04FC4 ITSO_SVC1 local
0000020061C06FCA ITSO_SVC4 remote   partially_configured_local  50
```

In Example 4-4, the partnership is created between `ITSO_SVC4` back to `ITSO_SVC1`, specifying the bandwidth to be used for a background copy of 50 MBps.

After creating the partnership, verify that the partnership is fully configured on both systems by reissuing the **`lspartnership`** command.

Example 4-4 Creating the partnership from ITSO_SVC4 to ITSO_SVC1 and verifying it

```
IBM_2145:ITSO_SVC4:admin>mkpartnership -bandwidth 50 ITSO_SVC1
IBM_2145:ITSO_SVC4:admin>lspartnership
id               name      location partnership      bandwidth
0000020061C06FCA ITSO_SVC4 local
000002006BE04FC4 ITSO_SVC1 remote   fully_configured 50
```

4.1.3 Creating Metro Mirror relationships

Here we illustrate how to create a Metro Mirror Consistency Group, add Metro Mirror relationships to the Consistency Group, and create a stand-alone Metro Mirror relationship.

Creating a Metro Mirror Consistency Group

In Example 4-5, we create the Metro Mirror Consistency Group using the **`mkrcconsistgrp`** command. This Consistency Group will be used for the Metro Mirror relationships of the database volumes named `MM_DB_Pri` and `MM_DBLog_Pri`. The Consistency Group is named `CG_W2K3_MM`.

Example 4-5 Creating the Metro Mirror Consistency Group CG_W2K3_MM

```
IBM_2145:ITSO_SVC1:admin>mkrcconsistgrp -cluster ITSO_SVC4 -name CG_W2K3_MM
RC Consistency Group, id [0], successfully created

IBM_2145:ITSO_SVC1:admin>lsrcconsistgrp
id name       master_cluster_id master_cluster_name aux_cluster_id
aux_cluster_name primary state relationship_count copy_type   cycling_mode
0  CG_W2K3_MM 000002006BE04FC4  ITSO_SVC1       0000020061C06FCA ITSO_SVC4
empty 0                 empty_group none
```

Adding Metro Mirror relationships to the Consistency Group

In Example 4-6, we create the Metro Mirror relationships `MMREL1` and `MMREL2` for `MM_DB_Pri` and `MM_DBLog_Pri`. We also make them members of the Metro Mirror Consistency Group `CG_W2K3_MM`. We use the **`lsvdisk`** command to list all of the volumes in the `ITSO_SVC1` system. We then use the **`lsrcrelationshipcandidate`** command to show the volumes in the `ITSO_SVC4` system.

By using this command, we check the possible candidates for `MM_DB_Pri`. After checking all of these conditions, we use the **`mkrcrelationship`** command to create the Metro Mirror relationship.

To verify the newly created Metro Mirror relationships, list them with the **`lsrcrelationship`** command.

Example 4-6 Creating Metro Mirror relationships MMREL1 and MMREL2

```
IBM_2145:ITSO_SVC1:admin>lsvdisk -filtervalue name=MM*
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 RC_change
0 MM_DB_Pri   0       io_grp0     online 0        Pool_DS3500-1  10.00GB  striped
6005076801AF813F1000000000000031 0         1        empty         0          0
no
1 MM_DBLog_Pri 0       io_grp0     online 0        Pool_DS3500-1  10.00GB  striped
6005076801AF813F1000000000000032 0         1        empty         0          0
no
2 MM_App_Pri  0       io_grp0     online 0        Pool_DS3500-1  10.00GB  striped
6005076801AF813F1000000000000033 0         1        empty         0          0
no
IBM_2145:ITSO_SVC1:admin>lsrcrelationshipcandidate
id vdisk_name
0  MM_DB_Pri
1  MM_DBLog_Pri
```

```
2  MM_App_Pri

IBM_2145:ITSO_SVC1:admin>lsrcrelationshipcandidate -aux ITSO_SVC4 -master MM_DB_Pri
id vdisk_name
0  MM_DB_Sec
1  MM_DBLog_Sec
2  MM_App_Sec

IBM_2145:ITSO_SVC1:admin>mkrcrelationship -master MM_DB_Pri -aux MM_DB_Sec -cluster ITSO_SVC4
-consistgrp CG_W2K3_MM -name MMREL1
RC Relationship, id [0], successfully created
IBM_2145:ITSO_SVC1:admin>mkrcrelationship -master MM_Log_Pri -aux MM_Log_Sec -cluster ITSO_SVC4
-consistgrp CG_W2K3_MM -name MMREL2
RC Relationship, id [3], successfully created

IBM_2145:ITSO_SVC1:admin>lsrcrelationship
id name   master_cluster_id master_cluster_name master_vdisk_id master_vdisk_name aux_cluster_id
aux_cluster_name aux_vdisk_id aux_vdisk_name primary consistency_group_id consistency_group_name
state                bg_copy_priority progress copy_type cycling_mode
0  MMREL1 000002006BE04FC4  ITSO_SVC1           0               MM_DB_Pri
0000020061C06FCA ITSO_SVC4        0            MM_DB_Sec      master  0
CG_W2K3_MM             inconsistent_stopped 50               0        metro     none
3  MMREL2 000002006BE04FC4  ITSO_SVC1           3               MM_Log_Pri
0000020061C06FCA ITSO_SVC4        3            MM_Log_Sec     master  0
CG_W2K3_MM             inconsistent_stopped 50               0        metro     none
```

Creating a stand-alone Metro Mirror relationship

In Example 4-7, we create the stand-alone Metro Mirror relationship `MMREL3` for `MM_App_Pri`. After it is created, we check the status of this Metro Mirror relationship.

Notice that the state of `MMREL3` is consistent_stopped. `MMREL3` is in this state, because it was created with the **`-sync`** option. The **`-sync`** option indicates that the secondary (auxiliary) volume is already synchronized with the primary (master) volume. Initial background synchronization is skipped when this option is used, even though the volumes are not actually synchronized in this scenario.

We want to illustrate the option of pre-synchronized master and auxiliary volumes, before setting up the relationship. We have created the new relationship for `MM_App_Sec` using the **`-sync`** option.

> **-sync option consideration:** The **`-sync`** option is only used when the target volume is already an *exact* mirror of the source volume.

`MMREL2` and `MMREL1` are in the inconsistent_stopped state, because they were not created with the **`-sync`** option. Therefore, their auxiliary volumes need to be synchronized with their primary volumes.

Example 4-7 Creating a stand-alone relationship and verifying it

```
IBM_2145:ITSO_SVC1:admin>mkrcrelationship -master MM_App_Pri -aux MM_App_Sec -sync
-cluster ITSO_SVC4 -name MMREL3
RC Relationship, id [2], successfully created

IBM_2145:ITSO_SVC1:admin>lsrcrelationship 2
id 2
```

```
name MMREL3
master_cluster_id 000002006BE04FC4
master_cluster_name ITSO_SVC1
master_vdisk_id 2
master_vdisk_name MM_App_Pri
aux_cluster_id 0000020061C06FCA
aux_cluster_name ITSO_SVC4
aux_vdisk_id 2
aux_vdisk_name MM_App_Sec
primary master
consistency_group_id
consistency_group_name
state consistent_stopped
bg_copy_priority 50
progress 100
freeze_time
status online
sync in_sync
copy_type metro
cycle_period_seconds 300
cycling_mode none
master_change_vdisk_id
master_change_vdisk_name
aux_change_vdisk_id
aux_change_vdisk_name
```

4.1.4 Starting Metro Mirror

Now that the Metro Mirror Consistency Group and relationships are in place, we are ready to use Metro Mirror relationships in our environment.

When implementing Metro Mirror, the goal is to reach a consistent and synchronized state that can provide redundancy for a data set if a failure occurs that affects the production site.

In the following section, we show how to stop and start stand-alone Metro Mirror relationships and Consistency Groups.

Starting a stand-alone Metro Mirror relationship

In Example 4-8, we start a stand-alone Metro Mirror relationship named `MMREL3`. Because the Metro Mirror relationship was in the Consistent stopped state and no updates have been made to the primary volume, the relationship quickly enters the Consistent synchronized state.

Example 4-8 Starting the stand-alone Metro Mirror relationship

```
IBM_2145:ITSO_SVC1:admin>startrcrelationship MMREL3
IBM_2145:ITSO_SVC1:admin>lsrcrelationship MMREL3
id 2
name MMREL3
master_cluster_id 000002006BE04FC4
master_cluster_name ITSO_SVC1
master_vdisk_id 2
master_vdisk_name MM_App_Pri
aux_cluster_id 0000020061C06FCA
aux_cluster_name ITSO_SVC4
```

```
aux_vdisk_id 2
aux_vdisk_name MM_App_Sec
primary master
consistency_group_id
consistency_group_name
state consistent_synchronized
bg_copy_priority 50
progress
freeze_time
status online
sync
copy_type metro
cycle_period_seconds 300
cycling_mode none
master_change_vdisk_id
master_change_vdisk_name
aux_change_vdisk_id
aux_change_vdisk_name
```

Starting a Metro Mirror Consistency Group

In Example 4-9, we start the Metro Mirror Consistency Group `CG_W2K3_MM`. Because the Consistency Group was in the Inconsistent stopped state, it enters the Inconsistent copying state until the background copy has completed for all of the relationships in the Consistency Group.

Upon completion of the background copy, it enters the Consistent synchronized state.

Example 4-9 Starting the Metro Mirror Consistency Group

```
IBM_2145:ITSO_SVC1:admin>startrcconsistgrp CG_W2K3_MM
IBM_2145:ITSO_SVC1:admin>lsrcconsistgrp
id name       master_cluster_id master_cluster_name aux_cluster_id
aux_cluster_name primary state                relationship_count copy_type
cycling_mode
0  CG_W2K3_MM 000002006BE04FC4  ITSO_SVC1         0000020061C06FCA ITSO_SVC4
master  inconsistent_copying 2                  metro     none
```

4.1.5 Monitoring the background copy progress

To monitor the background copy progress, we can use the **`lsrcrelationship`** command. This command shows all of the defined Metro Mirror relationships if it is used without any arguments. In the command output, `progress` indicates the current background copy progress. Our Metro Mirror relationship is shown in Example 4-10.

> **Tip:** Setting up SNMP traps for the SVC enables automatic notification when Metro Mirror Consistency Groups or relationships change state.

Example 4-10 Monitoring the background copy progress example

```
IBM_2145:ITSO_SVC1:admin>lsrcrelationship MMREL1
id 0
name MMREL1
master_cluster_id 000002006BE04FC4
master_cluster_name ITSO_SVC1
```

```
master_vdisk_id 0
master_vdisk_name MM_DB_Pri
aux_cluster_id 0000020061C06FCA
aux_cluster_name ITSO_SVC4
aux_vdisk_id 0
aux_vdisk_name MM_DB_Sec
primary master
consistency_group_id 0
consistency_group_name CG_W2K3_MM
state inconsistent_copying
bg_copy_priority 50
progress 81
freeze_time
status online
sync
copy_type metro
cycle_period_seconds 300
cycling_mode none
master_change_vdisk_id
master_change_vdisk_name
aux_change_vdisk_id
aux_change_vdisk_name

IBM_2145:ITSO_SVC1:admin>lsrcrelationship MMREL2
id 3
name MMREL2
master_cluster_id 000002006BE04FC4
master_cluster_name ITSO_SVC1
master_vdisk_id 3
master_vdisk_name MM_Log_Pri
aux_cluster_id 0000020061C06FCA
aux_cluster_name ITSO_SVC4
aux_vdisk_id 3
aux_vdisk_name MM_Log_Sec
primary master
consistency_group_id 0
consistency_group_name CG_W2K3_MM
state inconsistent_copying
bg_copy_priority 50
progress 82
freeze_time
status online
sync
copy_type metro
cycle_period_seconds 300
cycling_mode none
master_change_vdisk_id
master_change_vdisk_name
aux_change_vdisk_id
aux_change_vdisk_name
```

When all Metro Mirror relationships have completed the background copy, the Consistency Group enters the Consistent synchronized state, as shown in Example 4-11 on page 86.

Example 4-11 Listing the Metro Mirror Consistency Group

```
IBM_2145:ITSO_SVC1:admin>lsrcconsistgrp CG_W2K3_MM
id 0
name CG_W2K3_MM
master_cluster_id 000002006BE04FC4
master_cluster_name ITSO_SVC1
aux_cluster_id 0000020061C06FCA
aux_cluster_name ITSO_SVC4
primary master
state consistent_synchronized
relationship_count 2
freeze_time
status
sync
copy_type metro
cycle_period_seconds 300
cycling_mode none
RC_rel_id 0
RC_rel_name MMREL1
RC_rel_id 3
RC_rel_name MMREL2
```

4.1.6 Stopping and restarting Metro Mirror

Now that the Metro Mirror Consistency Group and relationships are running, in this section and in the following sections we describe how to stop, restart, and change the direction of the stand-alone Metro Mirror relationships and the Consistency Group.

Stopping a stand-alone Metro Mirror relationship

Example 4-12 shows how to stop the stand-alone Metro Mirror relationship, while enabling access (write I/O) to both the primary and secondary volumes. It also shows the relationship entering the Idling state.

Example 4-12 Stopping stand-alone Metro Mirror relationship and enabling access to the secondary

```
IBM_2145:ITSO_SVC1:admin>stoprcrelationship -access MMREL3
IBM_2145:ITSO_SVC1:admin>lsrcrelationship MMREL3
id 2
name MMREL3
master_cluster_id 000002006BE04FC4
master_cluster_name ITSO_SVC1
master_vdisk_id 2
master_vdisk_name MM_App_Pri
aux_cluster_id 0000020061C06FCA
aux_cluster_name ITSO_SVC4
aux_vdisk_id 2
aux_vdisk_name MM_App_Sec
primary
consistency_group_id
consistency_group_name
state idling
bg_copy_priority 50
progress
freeze_time
```

```
status
sync in_sync
copy_type metro
cycle_period_seconds 300
cycling_mode none
master_change_vdisk_id
master_change_vdisk_name
aux_change_vdisk_id
aux_change_vdisk_name
```

Stopping a Metro Mirror Consistency Group

Example 4-13 shows how to stop the Metro Mirror Consistency Group without specifying the **-access** flag. The Consistency Group enters the Consistent stopped state.

Example 4-13 Stopping a Metro Mirror Consistency Group

```
IBM_2145:ITSO_SVC1:admin>stoprcconsistgrp CG_W2K3_MM
IBM_2145:ITSO_SVC1:admin>lsrcconsistgrp CG_W2K3_MM
id 0
name CG_W2K3_MM
master_cluster_id 000002006BE04FC4
master_cluster_name ITSO_SVC1
aux_cluster_id 0000020061C06FCA
aux_cluster_name ITSO_SVC4
primary master
state consistent_stopped
relationship_count 2
freeze_time
status
sync in_sync
copy_type metro
cycle_period_seconds 300
cycling_mode none
RC_rel_id 0
RC_rel_name MMREL1
RC_rel_id 3
RC_rel_name MMREL2
```

If we later want to enable access (write I/O) to the secondary volume, we reissue the **stoprcconsistgrp** command, specifying the **-access** flag. The Consistency Group transits to the Idling state, as shown in Example 4-14.

Example 4-14 Stopping a Metro Mirror Consistency Group and enabling access to the secondary

```
IBM_2145:ITSO_SVC1:admin>stoprcconsistgrp -access CG_W2K3_MM
IBM_2145:ITSO_SVC1:admin>lsrcconsistgrp CG_W2K3_MM
id 0
name CG_W2K3_MM
master_cluster_id 000002006BE04FC4
master_cluster_name ITSO_SVC1
aux_cluster_id 0000020061C06FCA
aux_cluster_name ITSO_SVC4
primary
state idling
relationship_count 2
```

```
freeze_time
status
sync in_sync
copy_type metro
cycle_period_seconds 300
cycling_mode none
RC_rel_id 0
RC_rel_name MMREL1
RC_rel_id 3
RC_rel_name MMREL2
```

Restarting a Metro Mirror relationship in the Idling state

When restarting a Metro Mirror relationship in the Idling state, we must specify the copy direction.

If any updates have been performed on either the master or the auxiliary volume, consistency will be compromised. Therefore, we must issue the command with the **-force** flag to restart a relationship, as shown in Example 4-15.

Example 4-15 Restarting a Metro Mirror relationship after updates in the Idling state

```
IBM_2145:ITSO_SVC1:admin>startrcrelationship -primary master -force MMREL3
IBM_2145:ITSO_SVC1:admin>lsrcrelationship MMREL3
id 2
name MMREL3
master_cluster_id 000002006BE04FC4
master_cluster_name ITSO_SVC1
master_vdisk_id 2
master_vdisk_name MM_App_Pri
aux_cluster_id 0000020061C06FCA
aux_cluster_name ITSO_SVC4
aux_vdisk_id 2
aux_vdisk_name MM_App_Sec
primary master
consistency_group_id
consistency_group_name
state consistent_synchronized
bg_copy_priority 50
progress
freeze_time
status online
sync
copy_type metro
cycle_period_seconds 300
cycling_mode none
master_change_vdisk_id
master_change_vdisk_name
aux_change_vdisk_id
aux_change_vdisk_name
```

Restarting a Metro Mirror Consistency Group in the Idling state

When restarting a Metro Mirror Consistency Group in the Idling state, we must specify the copy direction.

If any updates have been performed on either the master or the auxiliary volume in any of the Metro Mirror relationships in the Consistency Group, the consistency is compromised. Therefore, we must use the **-force** flag to start a relationship. If the **-force** flag is not used, the command fails.

In Example 4-16, we change the copy direction by specifying the auxiliary volumes to become the primaries.

Example 4-16 Restarting a Metro Mirror relationship while changing the copy direction

```
IBM_2145:ITSO_SVC1:admin>startrcconsistgrp -force -primary aux CG_W2K3_MM
IBM_2145:ITSO_SVC1:admin>lsrcconsistgrp CG_W2K3_MM
id 0
name CG_W2K3_MM
master_cluster_id 000002006BE04FC4
master_cluster_name ITSO_SVC1
aux_cluster_id 0000020061C06FCA
aux_cluster_name ITSO_SVC4
primary aux
state consistent_synchronized
relationship_count 2
freeze_time
status
sync
copy_type metro
cycle_period_seconds 300
cycling_mode none
RC_rel_id 0
RC_rel_name MMREL1
RC_rel_id 3
RC_rel_name MMREL2
```

4.1.7 Changing the copy direction on Metro Mirror

In this section, we show how to change the copy direction of the stand-alone Metro Mirror volume relationship and of the Consistency Group.

Switching the copy direction on a Metro Mirror volume relationship

When a Metro Mirror relationship is in the Consistent synchronized state, you can change the copy direction for the relationship using the **switchrcrelationship** command, specifying the primary volume. If the specified volume is already a primary when you issue this command, the command has no effect. The **switchrcrelationship** command is issued to reverse the roles of the primary and secondary volume as part of a failover process during a disaster recovery event.

In Example 4-17 on page 90, we change the copy direction for the stand-alone Metro Mirror relationship by specifying the auxiliary volume to become the primary.

> **Host write activity consideration:** Ensure there is no outstanding host write activity on the volumes that are about to transition from a primary role to a secondary role.

Example 4-17 Switching the copy direction for a Metro Mirror Consistency Group

```
IBM_2145:ITSO_SVC1:admin>lsrcrelationship MMREL3
id 2
name MMREL3
master_cluster_id 000002006BE04FC4
master_cluster_name ITSO_SVC1
master_vdisk_id 2
master_vdisk_name MM_App_Pri
aux_cluster_id 0000020061C06FCA
aux_cluster_name ITSO_SVC4
aux_vdisk_id 2
aux_vdisk_name MM_App_Sec
primary master
consistency_group_id
consistency_group_name
state consistent_synchronized
bg_copy_priority 50
progress
freeze_time
status online
sync
copy_type metro
cycle_period_seconds 300
cycling_mode none
master_change_vdisk_id
master_change_vdisk_name
aux_change_vdisk_id
aux_change_vdisk_name
IBM_2145:ITSO_SVC1:admin>switchrcrelationship -primary aux MMREL3
IBM_2145:ITSO_SVC1:admin>lsrcrelationship MMREL3
id 2
name MMREL3
master_cluster_id 000002006BE04FC4
master_cluster_name ITSO_SVC1
master_vdisk_id 2
master_vdisk_name MM_App_Pri
aux_cluster_id 0000020061C06FCA
aux_cluster_name ITSO_SVC4
aux_vdisk_id 2
aux_vdisk_name MM_App_Sec
primary aux
consistency_group_id
consistency_group_name
state consistent_synchronized
bg_copy_priority 50
progress
freeze_time
status online
sync
copy_type metro
cycle_period_seconds 300
cycling_mode none
master_change_vdisk_id
master_change_vdisk_name
aux_change_vdisk_id
aux_change_vdisk_name
```

Switching the copy direction on a Metro Mirror Consistency Group

When a Metro Mirror Consistency Group is in the Consistent synchronized state, you can change the copy direction for the Consistency Group by using the **switchrcconsistgrp** command and specifying the primary volume. If the specified volume is already a primary when you issue this command, the command has no effect.

In Example 4-18, we change the copy direction for the Metro Mirror Consistency Group by specifying the auxiliary volume to become the primary volume.

> **Host write activity consideration:** Ensure there is no outstanding host write activity on the volumes that are about to transition from a primary role to a secondary role.

Example 4-18 Switching the copy direction for a Metro Mirror Consistency Group

```
IBM_2145:ITSO_SVC1:admin>lsrcconsistgrp CG_W2K3_MM
id 0
name CG_W2K3_MM
master_cluster_id 000002006BE04FC4
master_cluster_name ITSO_SVC1
aux_cluster_id 0000020061C06FCA
aux_cluster_name ITSO_SVC4
primary master
state consistent_synchronized
relationship_count 2
freeze_time
status
sync
copy_type metro
cycle_period_seconds 300
cycling_mode none
RC_rel_id 0
RC_rel_name MMREL1
RC_rel_id 3
RC_rel_name MMREL2
IBM_2145:ITSO_SVC1:admin>switchrcconsistgrp -primary aux CG_W2K3_MM
IBM_2145:ITSO_SVC1:admin>lsrcconsistgrp CG_W2K3_MM
id 0
name CG_W2K3_MM
master_cluster_id 000002006BE04FC4
master_cluster_name ITSO_SVC1
aux_cluster_id 0000020061C06FCA
aux_cluster_name ITSO_SVC4
primary aux
state consistent_synchronized
relationship_count 2
freeze_time
status
sync
copy_type metro
cycle_period_seconds 300
cycling_mode none
RC_rel_id 0
RC_rel_name MMREL1
RC_rel_id 3
RC_rel_name MMREL2
```

4.1.8 Creating multiple partnerships

Starting with SVC 5.1, you can have a clustered system partnership among up to four systems in a variety of topologies, as listed here:

- Star
- Ring
- Mesh
- Linear

In this section, we describe how to configure the system partnership for each configuration.

In our scenarios, we configure the partnership by referring to the clustered systems as A, B, C, and D:

- `ITSO_SVC1` = A
- `ITSO_SVC2` = B
- `ITSO_SVC3` = C
- `ITSO_SVC4` = D

Example 4-19 shows the available systems for a partnership using the **`lsclustercandidate`** command on each system.

Example 4-19 Available clustered systems

```
IBM_2145:ITSO_SVC1:admin>lspartnershipcandidate
id               configured name
0000020061C06FCA no         ITSO_SVC4
0000020060A06FB8 no         ITSO_SVC3
000002006AC03A42 no         ITSO_SVC2

IBM_2145:ITSO_SVC2:admin>lspartnershipcandidate
id               configured name
0000020061C06FCA no         ITSO_SVC4
000002006BE04FC4 no         ITSO_SVC1
0000020060A06FB8 no         ITSO_SVC3

IBM_2145:ITSO_SVC3:admin>lspartnershipcandidate
id               configured name
000002006BE04FC4 no         ITSO_SVC1
0000020061C06FCA no         ITSO_SVC4
000002006AC03A42 no         ITSO_SVC2

IBM_2145:ITSO_SVC4:admin>lspartnershipcandidate
id               configured name
000002006BE04FC4 no         ITSO_SVC1
0000020060A06FB8 no         ITSO_SVC3
000002006AC03A42 no         ITSO_SVC2
```

Partnerships in a star topology

Figure 4-2 on page 93 shows the star configuration.

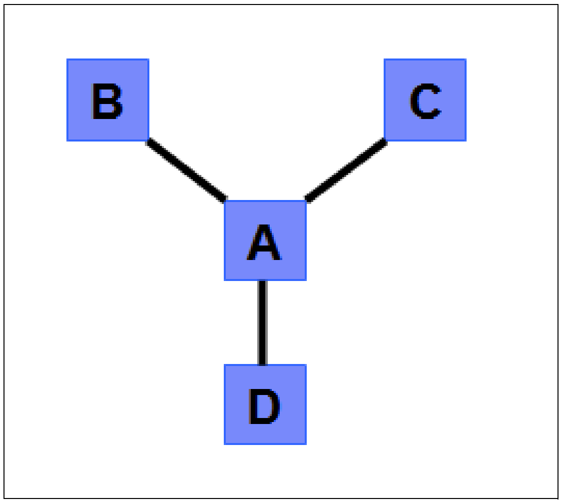


Figure 4-2 Star topology

Example 4-20 shows the sequence of **mkpartnership** commands to execute to create a star configuration.

Example 4-20 Creating a star configuration using the mkpartnership command

```
From ITSO_SVC1 to multiple systems

IBM_2145:ITSO_SVC1:admin>mkpartnership -bandwidth 50 ITSO_SVC2
IBM_2145:ITSO_SVC1:admin>mkpartnership -bandwidth 50 ITSO_SVC3
IBM_2145:ITSO_SVC1:admin>mkpartnership -bandwidth 50 ITSO_SVC4

From ITSO_SVC2 to ITSO_SVC1

IBM_2145:ITSO_SVC2:admin>mkpartnership -bandwidth 50 ITSO_SVC1

From ITSO_SVC3 to ITSO_SVC1

IBM_2145:ITSO_SVC3:admin>mkpartnership -bandwidth 50 ITSO_SVC1

From ITSO_SVC4 to ITSO_SVC1

IBM_2145:ITSO_SVC4:admin>mkpartnership -bandwidth 50 ITSO_SVC1

From ITSO_SVC1

IBM_2145:ITSO_SVC1:admin>lspartnership
id               name      location partnership      bandwidth
000002006BE04FC4 ITSO_SVC1 local
000002006AC03A42 ITSO_SVC2 remote   fully_configured 50
0000020060A06FB8 ITSO_SVC3 remote   fully_configured 50
0000020061C06FCA ITSO_SVC4 remote   fully_configured 50

From ITSO_SVC2
```

```
IBM_2145:ITSO_SVC2:admin>lspartnership
id               name                 location partnership
bandwidth
000002006AC03A42 ITSO_SVC2            local
000002006BE04FC4 ITSO_SVC1            remote   fully_configured         50
```

From ITSO_SVC3

```
IBM_2145:ITSO_SVC3:admin>lspartnership
id               name      location partnership      bandwidth
0000020060A06FB8 ITSO_SVC3 local
000002006BE04FC4 ITSO_SVC1 remote   fully_configured 50
```

From ITSO_SVC4

```
IBM_2145:ITSO_SVC4:admin>lspartnership
id               name      location partnership      bandwidth
0000020061C06FCA ITSO_SVC4 local
000002006BE04FC4 ITSO_SVC1 remote   fully_configured 50
```

After the system partnership has been configured, you can configure relationships and Consistency Groups.

Partnerships in a ring topology

Figure 4-3 shows a ring topology. A triangle is the simplest form of a ring. A square is also a supported ring topology.

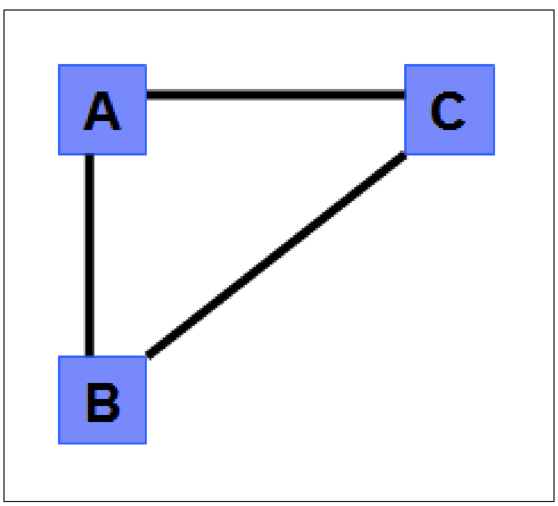


Figure 4-3 Ring topology

Example 4-21 shows the sequence of **mkpartnership** commands to execute to create a triangle configuration.

Example 4-21 Creating a triangle configuration

From ITSO_SVC1 to ITSO_SVC2 and ITSO_SVC3

```
IBM_2145:ITSO_SVC1:admin>mkpartnership -bandwidth 50 ITSO_SVC2
IBM_2145:ITSO_SVC1:admin>mkpartnership -bandwidth 50 ITSO_SVC3
IBM_2145:ITSO_SVC1:admin>lspartnership
```

```
id               name      location partnership               bandwidth
000002006BE04FC4 ITSO_SVC1 local
000002006AC03A42 ITSO_SVC2 remote   partially_configured_local 50
0000020060A06FB8 ITSO_SVC3 remote   partially_configured_local 50
```

From ITSO_SVC2 to ITSO_SVC1 and ITSO_SVC3

```
IBM_2145:ITSO_SVC2:admin>mkpartnership -bandwidth 50 ITSO_SVC1
IBM_2145:ITSO_SVC2:admin>mkpartnership -bandwidth 50 ITSO_SVC3
IBM_2145:ITSO_SVC2:admin>lspartnership
id               name                  location partnership
bandwidth
000002006AC03A42 ITSO_SVC2             local
000002006BE04FC4 ITSO_SVC1             remote   fully_configured           50
0000020060A06FB8 ITSO_SVC3             remote   partially_configured_local 50
```

From ITSO_SVC3 to ITSO_SVC1 and ITSO_SVC2

```
IBM_2145:ITSO_SVC3:admin>mkpartnership -bandwidth 50 ITSO_SVC1
IBM_2145:ITSO_SVC3:admin>mkpartnership -bandwidth 50 ITSO_SVC2
IBM_2145:ITSO_SVC3:admin>lspartnership
id               name      location partnership     bandwidth
0000020060A06FB8 ITSO_SVC3 local
000002006BE04FC4 ITSO_SVC1 remote   fully_configured 50
000002006AC03A42 ITSO_SVC2 remote   fully_configured 50
```

After the system partnership has been configured, you can configure any relationships or Consistency Groups that you need.

Partnerships in a fully connected mesh topology

Figure 4-4 shows a fully connected mesh configuration. A mesh does not have to be fully connected; for example, you can leave out the A to D connection.

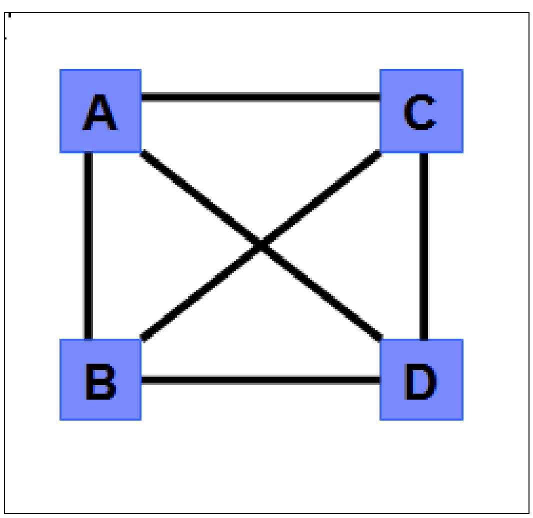


Figure 4-4 Mesh (fully connected) configuration

Example 4-22 on page 96 shows the sequence of **mkpartnership** commands to execute to create a fully connected configuration.

Example 4-22 Creating a fully connected configuration

```
From ITSO_SVC1 to ITSO_SVC2, ITSO_SVC3 and ITSO_SVC4

IBM_2145:ITSO_SVC1:admin>mkpartnership -bandwidth 50 ITSO_SVC2
IBM_2145:ITSO_SVC1:admin>mkpartnership -bandwidth 50 ITSO_SVC3
IBM_2145:ITSO_SVC1:admin>mkpartnership -bandwidth 50 ITSO_SVC4
IBM_2145:ITSO_SVC1:admin>lspartnership
id               name      location partnership                bandwidth
000002006BE04FC4 ITSO_SVC1 local
000002006AC03A42 ITSO_SVC2 remote   partially_configured_local 50
0000020060A06FB8 ITSO_SVC3 remote   partially_configured_local 50
0000020061C06FCA ITSO_SVC4 remote   partially_configured_local 50

From ITSO_SVC2 to ITSO_SVC1, ITSO_SVC3 and ITSO-SVC4

IBM_2145:ITSO_SVC2:admin>mkpartnership -bandwidth 50 ITSO_SVC1
IBM_2145:ITSO_SVC2:admin>mkpartnership -bandwidth 50 ITSO_SVC3
IBM_2145:ITSO_SVC2:admin>mkpartnership -bandwidth 50 ITSO_SVC4
IBM_2145:ITSO_SVC2:admin>lspartnership
id               name      location partnership                bandwidth
000002006AC03A42 ITSO_SVC2 local
000002006BE04FC4 ITSO_SVC1 remote   fully_configured           50
0000020060A06FB8 ITSO_SVC3 remote   partially_configured_local 50
0000020061C06FCA ITSO_SVC4 remote   partially_configured_local 50

From ITSO_SVC3 to ITSO_SVC1, ITSO_SVC3 and ITSO-SVC4

IBM_2145:ITSO_SVC3:admin>mkpartnership -bandwidth 50 ITSO_SVC1
IBM_2145:ITSO_SVC3:admin>mkpartnership -bandwidth 50 ITSO_SVC2
IBM_2145:ITSO_SVC3:admin>mkpartnership -bandwidth 50 ITSO_SVC4
IBM_2145:ITSO_SVC3:admin>lspartnership
id               name      location partnership                bandwidth
0000020060A06FB8 ITSO_SVC3 local
000002006BE04FC4 ITSO_SVC1 remote   fully_configured           50
000002006AC03A42 ITSO_SVC2 remote   fully_configured           50
0000020061C06FCA ITSO_SVC4 remote   partially_configured_local 50

From ITSO-SVC4 to ITSO_SVC1, ITSO_SVC2 and ITSO_SVC3

IBM_2145:ITSO_SVC4:admin>mkpartnership -bandwidth 50 ITSO_SVC1
IBM_2145:ITSO_SVC4:admin>mkpartnership -bandwidth 50 ITSO_SVC2
IBM_2145:ITSO_SVC4:admin>mkpartnership -bandwidth 50 ITSO_SVC3
IBM_2145:ITSO_SVC4:admin>lspartnership
id               name      location partnership      bandwidth
0000020061C06FCA ITSO_SVC4 local
000002006BE04FC4 ITSO_SVC1 remote   fully_configured 50
000002006AC03A42 ITSO_SVC2 remote   fully_configured 50
0000020060A06FB8 ITSO_SVC3 remote   fully_configured 50
```

After the system partnership has been configured, you can configure any relationships or Consistency Groups that you need.

Partnerships in a linear topology

Figure 4-5 on page 97 shows a linear topology.

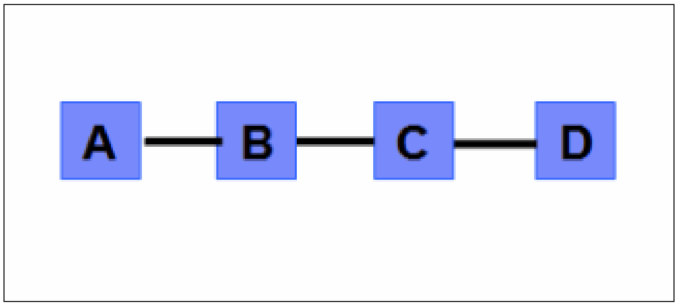


Figure 4-5 Linear configuration

Example 4-23 shows the sequence of **mkpartnership** commands to execute to create a linear configuration.

Example 4-23 Creating a linear configuration

```
From ITSO_SVC1 to ITSO_SVC2

IBM_2145:ITSO_SVC1:admin>mkpartnership -bandwidth 50 ITSO_SVC2
IBM_2145:ITSO_SVC1:admin>lspartnership
id               name      location partnership               bandwidth
000002006BE04FC4 ITSO_SVC1 local
000002006AC03A42 ITSO_SVC2 remote   partially_configured_local 50

From ITSO_SVC2 to ITSO_SVC1 and ITSO_SVC3

IBM_2145:ITSO_SVC2:admin>mkpartnership -bandwidth 50 ITSO_SVC1
IBM_2145:ITSO_SVC2:admin>mkpartnership -bandwidth 50 ITSO_SVC3
IBM_2145:ITSO_SVC2:admin>lspartnership
id               name      location partnership               bandwidth
000002006AC03A42 ITSO_SVC2 local
000002006BE04FC4 ITSO_SVC1 remote   fully_configured           50
0000020060A06FB8 ITSO_SVC3 remote   partially_configured_local 50

From ITSO_SVC3 to ITSO_SVC2 and ITSO_SVC4

IBM_2145:ITSO_SVC3:admin>mkpartnership -bandwidth 50 ITSO_SVC2
IBM_2145:ITSO_SVC3:admin>mkpartnership -bandwidth 50 ITSO_SVC4
IBM_2145:ITSO_SVC3:admin>lspartnership
id               name      location partnership               bandwidth
0000020060A06FB8 ITSO_SVC3 local
000002006AC03A42 ITSO_SVC2 remote   fully_configured           50
0000020061C06FCA ITSO_SVC4 remote   partially_configured_local 50

From ITSO_SVC4 to ITSO_SVC3

IBM_2145:ITSO_SVC4:admin>mkpartnership -bandwidth 50 ITSO_SVC3
IBM_2145:ITSO_SVC4:admin>lspartnership
id               name      location partnership      bandwidth
0000020061C06FCA ITSO_SVC4 local
0000020060A06FB8 ITSO_SVC3 remote   fully_configured 50
```

After the system partnership has been configured, you can configure any relationships or Consistency Groups that you need.

4.2 Global Mirror using the CLI

In the following scenario, we set up an inter-cluster Global Mirror relationship between the SVC system `ITSO_SVC1` at the primary site and the SVC system `ITSO_SVC4` at the secondary site.

> **Cluster consideration:** This example is for an inter-cluster Global Mirror partnership.
>
> If you want to set up an *intra-cluster* operation, we highlight those parts in the following procedure that you do *not* need to perform.

Table 4-2 shows the details of the volumes.

Table 4-2 Details of volumes for Global Mirror relationship scenario

Content of volume	Volumes at primary site	Volumes at secondary site
Database files	GM_DB_Pri	GM_DB_Sec
Database log files	GM_DBLog_Pri	GM_DBLog_Sec
Application files	GM_App_Pri	GM_App_Sec

Because data consistency is needed across `GM_DB_Pri` and `GM_DBLog_Pri`, we create a Consistency Group to handle Global Mirror relationships for them.

Because in this scenario the application files are independent of the database, we create a stand-alone Global Mirror relationship for `GM_App_Pri`. Figure 4-6 illustrates the Global Mirror relationship setup.

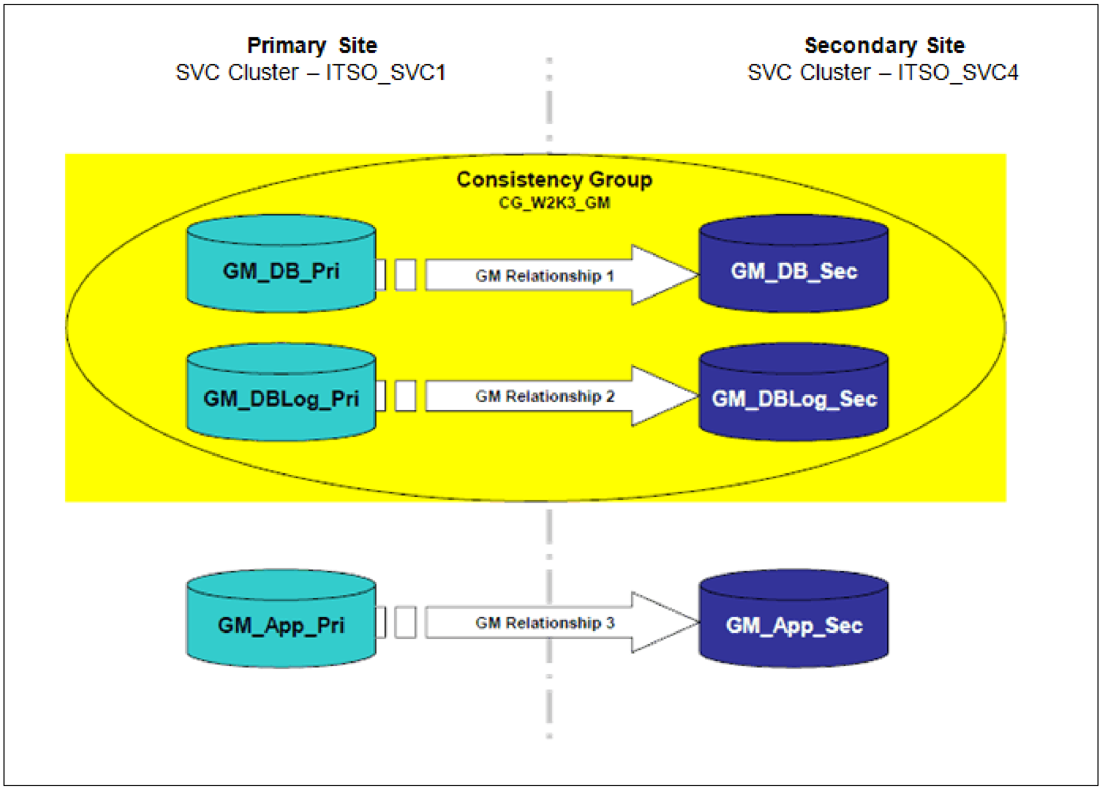


Figure 4-6 Global Mirror scenario

4.2.1 Setting up Global Mirror

In the following section, we assume that the source and target volumes have already been created and that the ISLs and zoning are in place, enabling the SVC systems to communicate.

To set up the Global Mirror, perform the following steps:

1. Create an SVC partnership between `ITSO_SVC1` and `ITSO_SVC4` on both SVC clustered systems.

   Bandwidth: `100` MBps

2. Create a Global Mirror Consistency Group.

   Name: `CG_W2K3_GM`

3. Create the Global Mirror relationship for `GM_DB_Pri`:

   – Master: `GM_DB_Pri`
   – Auxiliary: `GM_DB_Sec`
   – Auxiliary SVC system: `ITSO_SVC4`
   – Name: `GMREL1`
   – Consistency Group: `CG_W2K3_GM`

4. Create the Global Mirror relationship for `GM_DBLog_Pri`:

   – Master: `GM_DBLog_Pri`
   – Auxiliary: `GM_DBLog_Sec`
   – Auxiliary SVC system: `ITSO_SVC4`
   – Name: `GMREL2`
   – Consistency Group: `CG_W2K3_GM`

5. Create the Global Mirror relationship for `GM_App_Pri`:
   – Master: `GM_App_Pri`
   – Auxiliary: `GM_App_Sec`
   – Auxiliary SVC system: `ITSO_SVC4`
   – Name: `GMREL3`

In the following sections, we perform each step by using the CLI.

4.2.2 Creating a partnership - Global Mirror

We create an SVC partnership between these clustered systems.

> **Intra-cluster consideration:** If you are creating an intra-cluster Global Mirror, do *not* perform the next step.
>
> Instead, go to 4.2.3, "Changing link tolerance and system delay simulation" on page 101.

Verifying network visibility

To verify that both clustered systems can communicate with each other, use the **`lspartnership`** command. Example 4-24 on page 100 confirms that our clustered systems are communicating, because `ITSO_SVC4` is an eligible SVC system candidate at `ITSO_SVC1` for the SVC system partnership, and vice versa. Therefore, both systems communicate with each other.

Example 4-24 Listing the available SVC systems for partnership

```
IBM_2145:ITSO_SVC1:admin>lspartnershipcandidate
id               configured name
0000020061C06FCA no         ITSO_SVC4
IBM_2145:ITSO_SVC4:admin>lspartnershipcandidate
id               configured name
000002006BE04FC4 no         ITSO_SVC1
```

In Example 4-25, we show the output of the **`lspartnership`** command before setting up the SVC systems' partnership for Global Mirror. We show this output for comparison after we have set up the SVC partnership.

Example 4-25 Pre-verification of system configuration

```
IBM_2145:ITSO_SVC1:admin>lspartnership
id               name      location partnership bandwidth
000002006BE04FC4 ITSO_SVC1 local

IBM_2145:ITSO_SVC4:admin>lspartnership
id               name      location partnership bandwidth
0000020061C06FCA ITSO_SVC4 local
```

Establishing the partnership

In Example 4-26, we create the partnership from `ITSO_SVC1` to `ITSO_SVC4`, specifying a 100 MBps bandwidth to use for the background copy. To verify the status of the newly created partnership, we issue the **`lspartnership`** command. Notice that the new partnership is only *partially* configured. It will remain partially configured until we run the **`mkpartnership`** command on the other clustered system.

Example 4-26 Creating the partnership from ITSO_SVC1 to ITSO_SVC4 and verifying it

```
IBM_2145:ITSO_SVC1:admin>mkpartnership -bandwidth 100 ITSO_SVC4
IBM_2145:ITSO_SVC1:admin>lspartnership
id               name      location partnership                bandwidth
000002006BE04FC4 ITSO_SVC1 local
0000020061C06FCA ITSO_SVC4 remote   partially_configured_local 100
```

In Example 4-27, we create the partnership from `ITSO_SVC4` back to `ITSO_SVC1`, specifying a 100 MBps bandwidth to be used for the background copy. After creating the partnership, verify that the partnership is *fully* configured by reissuing the **`lspartnership`** command.

Example 4-27 Creating the partnership from ITSO_SVC4 to ITSO_SVC1 and verifying it

```
IBM_2145:ITSO_SVC4:admin>mkpartnership -bandwidth 100 ITSO_SVC1

IBM_2145:ITSO_SVC4:admin>lspartnership
id               name      location partnership      bandwidth
0000020061C06FCA ITSO_SVC4 local
000002006BE04FC4 ITSO_SVC1 remote   fully_configured 100

IBM_2145:ITSO_SVC1:admin>lspartnership
id               name      location partnership      bandwidth
000002006BE04FC4 ITSO_SVC1 local
0000020061C06FCA ITSO_SVC4 remote   fully_configured 100
```

4.2.3 Changing link tolerance and system delay simulation

The **`gmlinktolerance`** parameter defines the sensitivity of the SVC to inter-link overload conditions. The value is the number of seconds of continuous link difficulties that will be tolerated before the SVC will stop the remote copy relationships to prevent affecting host I/O at the primary site. To change the value, use the following command:

```
chsystem -gmlinktolerance link_tolerance
```

The **`gmlinktolerance`** value is between 60 and 86,400 seconds in increments of 10 seconds. The default value for the link tolerance is 300 seconds. A value of `0` disables link tolerance.

> **gmlinktolerance consideration:** Thoroughly test any changes to **`gmlinktolerance`** and carefully monitor host impact before they are put into a live production environment. Unless you have a specific reason for changing this parameter, we suggest you use the default value. A value that is too low can impact Global Mirror relationships. A value that is too high can impact hosts.

Inter-cluster and intra-cluster delay simulation

This Global Mirror feature permits a simulation of a delayed write to a remote volume. This feature allows testing to be performed that detects colliding writes. You can use this feature to test an application before the full deployment of the Global Mirror feature. The delay simulation can be enabled separately for each intra-cluster or inter-cluster Global Mirror. To enable this feature, run the following command either for the intra-cluster or inter-cluster simulation:

- For inter-cluster:

  ```
  chsystem -gminterdelaysimulation <inter_cluster_delay_simulation>
  ```

- For intra-cluster:

  `chsystem -gmintradelaysimulation <intra_cluster_delay_simulation>`

The **gminterdelaysimulation** and **gmintradelaysimulation** parameters set the amount of time (in milliseconds) that secondary I/Os are delayed. The valid range is from 0 to 100 milliseconds. A value of 0 (zero) disables the feature.

To check the current settings for the delay simulation, use the following command:

```
lssystem
```

In Example 4-28, we show the modification of the delay simulation value and a change of the Global Mirror link tolerance parameters. We also show the changed values of the Global Mirror link tolerance and delay simulation parameters.

Example 4-28 Delay simulation and link tolerance modification

```
IBM_2145:ITSO_SVC1:admin>chsystem -gminterdelaysimulation 20
IBM_2145:ITSO_SVC1:admin>chsystem -gmintradelaysimulation 40
IBM_2145:ITSO_SVC1:admin>chsystem -gmlinktolerance 200
IBM_2145:ITSO_SVC1:admin>lssystem
id 000002006BE04FC4
name ITSO_SVC1
location local
partnership
bandwidth
total_mdisk_capacity 866.5GB
space_in_mdisk_grps 766.5GB
space_allocated_to_vdisks 30.00GB
total_free_space 836.5GB
total_vdiskcopy_capacity 30.00GB
total_used_capacity 30.00GB
total_overallocation 3
total_vdisk_capacity 30.00GB
total_allocated_extent_capacity 31.50GB
statistics_status on
statistics_frequency 15
cluster_locale en_US
time_zone 520 US/Pacific
code_level 6.3.0.0 (build 54.0.1109090000)
console_IP 10.18.228.81:443
id_alias 000002006BE04FC4
gm_link_tolerance 200
gm_inter_cluster_delay_simulation 20
gm_intra_cluster_delay_simulation 40
gm_max_host_delay 5
email_reply
email_contact
email_contact_primary
email_contact_alternate
email_contact_location
email_contact2
email_contact2_primary
email_contact2_alternate
email_state stopped
inventory_mail_interval 0
cluster_ntp_IP_address
```

```
cluster_isns_IP_address
iscsi_auth_method chap
iscsi_chap_secret passw0rd
auth_service_configured no
auth_service_enabled no
auth_service_url
auth_service_user_name
auth_service_pwd_set no
auth_service_cert_set no
auth_service_type tip
relationship_bandwidth_limit 25
tier generic_ssd
tier_capacity 0.00MB
tier_free_capacity 0.00MB
tier generic_hdd
tier_capacity 766.50GB
tier_free_capacity 736.50GB
has_nas_key no
layer appliance
```

4.2.4 Creating Global Mirror relationships

In this section we show you how to create a Global Mirror Consistency Group, add Global Mirror relationships to a Consistency Group, and create a stand-alone Global Mirror relationship.

Creating a Global Mirror Consistency Group

In Example 4-29, we create the Global Mirror Consistency Group using the **mkrcconsistgrp** command. We will use this Consistency Group for the Global Mirror relationships for the database volumes. The Consistency Group is named `CG_W2K3_GM`.

Example 4-29 Creating the Global Mirror Consistency Group CG_W2K3_GM

```
IBM_2145:ITSO_SVC1:admin>mkrcconsistgrp -cluster ITSO_SVC4 -name CG_W2K3_GM
RC Consistency Group, id [0], successfully created
IBM_2145:ITSO_SVC1:admin>lsrcconsistgrp
id name       master_cluster_id master_cluster_name aux_cluster_id
aux_cluster_name primary state relationship_count copy_type   cycling_mode
0  CG_W2K3_GM 000002006BE04FC4  ITSO_SVC1       0000020061C06FCA ITSO_SVC4
empty 0                  empty_group none
```

Adding Global Mirror relationships to the Consistency Group

In Example 4-30, we create the `GMREL1` and `GMREL2` Global Mirror relationships for the `GM_DB_Pri` and `GM_DBLog_Pri` volumes. We also make them members of the `CG_W2K3_GM` Global Mirror Consistency Group.

We use the **lsvdisk** command to list all of the volumes in the `ITSO_SVC1` system, and then use the **lsrcrelationshipcandidate** command to show the possible candidate volumes for `GM_DB_Pri` in `ITSO_SVC4`.

After checking all of these conditions, we use the **mkrcrelationship** command to create the Global Mirror relationship. To verify the newly created Global Mirror relationships, we list them with the **lsrcrelationship** command.

Example 4-30 Creating GMREL1 and GMREL2 Global Mirror relationships

```
IBM_2145:ITSO_SVC1:admin>lsvdisk -filtervalue name=GM*
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 RC_change
0 GM_DB_Pri   0       io_grp0    online 0       Pool_DS3500-1  10.00GB  striped
6005076801AF813F1000000000000031 0         1       empty         0          0
no
1 GM_DBLog_Pri 0       io_grp0    online 0       Pool_DS3500-1  10.00GB  striped
6005076801AF813F1000000000000032 0         1       empty         0          0
no
2 GM_App_Pri  0       io_grp0    online 0       Pool_DS3500-1  10.00GB  striped
6005076801AF813F1000000000000033 0         1       empty         0          0
no

IBM_2145:ITSO_SVC1:admin>lsrcrelationshipcandidate -aux ITSO_SVC4 -master GM_DB_Pri
id vdisk_name
0  GM_DB_Sec
1  GM_DBLog_Sec
2  GM_App_Sec

IBM_2145:ITSO_SVC1:admin>mkrcrelationship -master GM_DB_Pri -aux GM_DB_Sec -cluster ITSO_SVC4
-consistgrp CG_W2K3_GM -name GMREL1 -global
RC Relationship, id [0], successfully created

IBM_2145:ITSO_SVC1:admin>mkrcrelationship -master GM_DBLog_Pri -aux GM_DBLog_Sec -cluster
ITSO_SVC4 -consistgrp CG_W2K3_GM -name GMREL2 -global
RC Relationship, id [1], successfully created

IBM_2145:ITSO_SVC1:admin>mkrcrelationship -master GM_DB_Pri -aux GM_DB_Sec -cluster ITSO_SVC4
-consistgrp CG_W2K3_GM -name GMREL1 -global
RC Relationship, id [2], successfully created
IBM_2145:ITSO_SVC1:admin>mkrcrelationship -master GM_DBLog_Pri -aux GM_DBLog_Sec -cluster
ITSO_SVC4 -consistgrp CG_W2K3_GM -name GMREL2 -global
RC Relationship, id [3], successfully created
IBM_2145:ITSO_SVC1:admin>lsrcrelationship
id name   master_cluster_id master_cluster_name master_vdisk_id master_vdisk_name aux_cluster_id
aux_cluster_name aux_vdisk_id aux_vdisk_name primary consistency_group_id consistency_group_name
state                bg_copy_priority progress copy_type cycling_mode
0  GMREL1 000002006BE04FC4  ITSO_SVC1          0              GM_DB_Pri
0000020061C06FCA ITSO_SVC4       0           GM_DB_Sec      master  0
CG_W2K3_GM             inconsistent_stopped 50              0       global    none
1  GMREL2 000002006BE04FC4  ITSO_SVC1          1              GM_DBLog_Pri
0000020061C06FCA ITSO_SVC4       1           GM_DBLog_Sec   master  0
CG_W2K3_GM             inconsistent_stopped 50              0       global    none
```

Creating a stand-alone Global Mirror relationship

In Example 4-31, we create the stand-alone Global Mirror relationship `GMREL3` for `GM_App_Pri`. After it is created, we will check the status of each of our Global Mirror relationships.

Notice that the status of `GMREL3` is consistent_stopped, because it was created with the **`-sync`** option. The **`-sync`** option indicates that the secondary (auxiliary) volume is already synchronized with the primary (master) volume. The initial background synchronization is skipped when this option is used.

GMREL1 and GMREL2 are in the inconsistent_stopped state, because they were not created with the **-sync** option, so their auxiliary volumes need to be synchronized with their primary volumes.

Example 4-31 Creating a stand-alone Global Mirror relationship and verifying it

```
IBM_2145:ITSO_SVC1:admin>mkrcrelationship -master GM_App_Pri -aux GM_App_Sec -cluster ITSO_SVC4
-sync -name GMREL3 -global
RC Relationship, id [2], successfully created

IBM_2145:ITSO_SVC1:admin>lsrcrelationship -delim :
id:name:master_cluster_id:master_cluster_name:master_vdisk_id:master_vdisk_name:aux_cluster_id:a
ux_cluster_name:aux_vdisk_id:aux_vdisk_name:primary:consistency_group_id:consistency_group_name:
state:bg_copy_priority:progress:copy_type:cycling_mode
0:GMREL1:000002006BE04FC4:ITSO_SVC1:0:GM_DB_Pri:0000020061C06FCA:ITSO_SVC4:0:GM_DB_Sec:master:0:
CG_W2K3_GM:inconsistent_copying:50:73:global:none
1:GMREL2:000002006BE04FC4:ITSO_SVC1:1:GM_DBLog_Pri:0000020061C06FCA:ITSO_SVC4:1:GM_DBLog_Sec:mas
ter:0:CG_W2K3_GM:inconsistent_copying:50:75:global:none
2:GMREL3:000002006BE04FC4:ITSO_SVC1:2:GM_App_Pri:0000020061C06FCA:ITSO_SVC4:2:GM_App_Sec:master:
::consistent_stopped:50:100:global:none
```

4.2.5 Starting Global Mirror

Now that we have created the Global Mirror Consistency Group and relationships, we are ready to use the Global Mirror relationships in our environment.

When implementing Global Mirror, the goal is to reach a consistent and synchronized state that can provide redundancy if a hardware failure occurs that affects the SAN at the production site.

In this section, we show how to start the stand-alone Global Mirror relationships and the Consistency Group.

Starting a stand-alone Global Mirror relationship

In Example 4-32, we start the stand-alone Global Mirror relationship named GMREL3. Because the Global Mirror relationship was in the Consistent stopped state and no updates have been made to the primary volume, the relationship quickly enters the Consistent synchronized state.

Example 4-32 Starting the stand-alone Global Mirror relationship

```
IBM_2145:ITSO_SVC1:admin>startrcrelationship GMREL3
IBM_2145:ITSO_SVC1:admin>lsrcrelationship GMREL3
id 2
name GMREL3
master_cluster_id 000002006BE04FC4
master_cluster_name ITSO_SVC1
master_vdisk_id 2
master_vdisk_name GM_App_Pri
aux_cluster_id 0000020061C06FCA
aux_cluster_name ITSO_SVC4
aux_vdisk_id 2
aux_vdisk_name GM_App_Sec
primary master
consistency_group_id
```

```
consistency_group_name
state consistent_synchronized
bg_copy_priority 50
progress
freeze_time
status online
sync
copy_type global
cycle_period_seconds 300
cycling_mode none
master_change_vdisk_id
master_change_vdisk_name
aux_change_vdisk_id
aux_change_vdisk_name
```

Starting a Global Mirror Consistency Group

In Example 4-33 on page 106, we start the `CG_W2K3_GM` Global Mirror Consistency Group.

Because the Consistency Group was in the Inconsistent stopped state, it enters the Inconsistent copying state until the background copy has completed for all of the relationships that are in the Consistency Group.

Upon completion of the background copy, the `CG_W2K3_GM` Global Mirror Consistency Group enters the Consistent synchronized state.

Example 4-33 Starting the Global Mirror Consistency Group

```
IBM_2145:ITSO_SVC1:admin>startrcconsistgrp CG_W2K3_GM
IBM_2145:ITSO_SVC1:admin>lsrcconsistgrp 0
id 0
name CG_W2K3_GM
master_cluster_id 000002006BE04FC4
master_cluster_name ITSO_SVC1
aux_cluster_id 0000020061C06FCA
aux_cluster_name ITSO_SVC4
primary master
state inconsistent_copying
relationship_count 2
freeze_time
status
sync
copy_type global
cycle_period_seconds 300
cycling_mode none
RC_rel_id 0
RC_rel_name GMREL1
RC_rel_id 1
RC_rel_name GMREL2
```

4.2.6 Monitoring the background copy progress

To monitor the background copy progress, use the **lsrcrelationship** command. This command displays all of the defined Global Mirror relationships if it is used without any parameters. In the command output, `progress` indicates the current background copy progress. Example 4-34 shows our Global Mirror relationships.

Tip: Setting up SNMP traps for the SVC enables automatic notification when Global Mirror Consistency Groups or relationships change state.

Example 4-34 Monitoring background copy progress example

```
IBM_2145:ITSO_SVC1:admin>lsrcrelationship GMREL1
id 0
name GMREL1
master_cluster_id 000002006BE04FC4
master_cluster_name ITSO_SVC1
master_vdisk_id 0
master_vdisk_name GM_DB_Pri
aux_cluster_id 0000020061C06FCA
aux_cluster_name ITSO_SVC4
aux_vdisk_id 0
aux_vdisk_name GM_DB_Sec
primary master
consistency_group_id 0
consistency_group_name CG_W2K3_GM
state inconsistent_copying
bg_copy_priority 50
progress 38
freeze_time
status online
sync
copy_type global
cycle_period_seconds 300
cycling_mode none
master_change_vdisk_id
master_change_vdisk_name
aux_change_vdisk_id
aux_change_vdisk_name

IBM_2145:ITSO_SVC1:admin>lsrcrelationship GMREL2
id 1
name GMREL2
master_cluster_id 000002006BE04FC4
master_cluster_name ITSO_SVC1
master_vdisk_id 1
master_vdisk_name GM_DBLog_Pri
aux_cluster_id 0000020061C06FCA
aux_cluster_name ITSO_SVC4
aux_vdisk_id 1
aux_vdisk_name GM_DBLog_Sec
primary master
consistency_group_id 0
consistency_group_name CG_W2K3_GM
state inconsistent_copying
```

```
bg_copy_priority 50
progress 76
freeze_time
status online
sync
copy_type global
cycle_period_seconds 300
cycling_mode none
master_change_vdisk_id
master_change_vdisk_name
aux_change_vdisk_id
aux_change_vdisk_name
```

When all of the Global Mirror relationships complete the background copy, the Consistency Group enters the Consistent synchronized state, as shown in Example 4-35.

Example 4-35 Listing the Global Mirror Consistency Group

```
IBM_2145:ITSO_SVC1:admin>lsrcconsistgrp CG_W2K3_GM
id 0
name CG_W2K3_GM
master_cluster_id 000002006BE04FC4
master_cluster_name ITSO_SVC1
aux_cluster_id 0000020061C06FCA
aux_cluster_name ITSO_SVC4
primary master
state consistent_synchronized
relationship_count 2
freeze_time
status
sync
copy_type global
cycle_period_seconds 300
cycling_mode none
RC_rel_id 0
RC_rel_name GMREL1
RC_rel_id 1
RC_rel_name GMREL2
```

4.2.7 Stopping and restarting Global Mirror

Now that the Global Mirror Consistency Group and relationships are running, we describe how to stop, restart, and change the direction of the stand-alone Global Mirror relationships and the Consistency Group.

First, we show how to stop and restart the stand-alone Global Mirror relationships and the Consistency Group.

Stopping a stand-alone Global Mirror relationship

In Example 4-36, we stop the stand-alone Global Mirror relationship while enabling access (write I/O) to both the primary and the secondary volume. As a result, the relationship enters the Idling state.

Example 4-36 Stopping the stand-alone Global Mirror relationship

```
IBM_2145:ITSO_SVC1:admin>stoprcrelationship -access GMREL3
IBM_2145:ITSO_SVC1:admin>lsrcrelationship GMREL3
id 2
name GMREL3
master_cluster_id 000002006BE04FC4
master_cluster_name ITSO_SVC1
master_vdisk_id 2
master_vdisk_name GM_App_Pri
aux_cluster_id 0000020061C06FCA
aux_cluster_name ITSO_SVC4
aux_vdisk_id 2
aux_vdisk_name GM_App_Sec
primary
consistency_group_id
consistency_group_name
state idling
bg_copy_priority 50
progress
freeze_time
status
sync in_sync
copy_type global
cycle_period_seconds 300
cycling_mode none
master_change_vdisk_id
master_change_vdisk_name
aux_change_vdisk_id
aux_change_vdisk_name
```

Stopping a Global Mirror Consistency Group

In Example 4-37, we stop the Global Mirror Consistency Group *without* specifying the **-access** parameter. Therefore, the Consistency Group enters the Consistent stopped state.

Example 4-37 Stopping a Global Mirror Consistency Group without specifying -access

```
IBM_2145:ITSO_SVC1:admin>stoprcconsistgrp CG_W2K3_GM
IBM_2145:ITSO_SVC1:admin>lsrcconsistgrp  CG_W2K3_GM
id 0
name CG_W2K3_GM
master_cluster_id 000002006BE04FC4
master_cluster_name ITSO_SVC1
aux_cluster_id 0000020061C06FCA
aux_cluster_name ITSO_SVC4
primary master
state consistent_stopped
relationship_count 2
freeze_time
status
sync in_sync
copy_type global
cycle_period_seconds 300
cycling_mode none
RC_rel_id 0
RC_rel_name GMREL1
```

```
RC_rel_id 1
RC_rel_name GMREL2
```

If, afterwards, we want to enable access (write I/O) for the secondary volume, we can reissue the **stoprcconsistgrp** command specifying the **-access** parameter. The Consistency Group transits to the Idling state, as shown in Example 4-38.

Example 4-38 Stopping a Global Mirror Consistency Group

```
IBM_2145:ITSO_SVC1:admin>stoprcconsistgrp -access CG_W2K3_GM
IBM_2145:ITSO_SVC1:admin>lsrcconsistgrp CG_W2K3_GM
id 0
name CG_W2K3_GM
master_cluster_id 000002006BE04FC4
master_cluster_name ITSO_SVC1
aux_cluster_id 0000020061C06FCA
aux_cluster_name ITSO_SVC4
primary
state idling
relationship_count 2
freeze_time
status
sync in_sync
copy_type global
cycle_period_seconds 300
cycling_mode none
RC_rel_id 0
RC_rel_name GMREL1
RC_rel_id 1
RC_rel_name GMREL2
```

Restarting a Global Mirror relationship in the Idling state

When restarting a Global Mirror relationship in the Idling state, we must specify the copy direction.

If any updates have been performed on either the master or the auxiliary volume, consistency will be compromised. Therefore, we must issue the **-force** parameter to restart the relationship. If the **-force** parameter is not used, the command will fail, as shown in Example 4-39.

Example 4-39 Restarting a Global Mirror relationship after updates in the Idling state

```
IBM_2145:ITSO_SVC1:admin>startrcrelationship -primary master -force GMREL3
IBM_2145:ITSO_SVC1:admin>lsrcrelationship GMREL3
id 2
name GMREL3
master_cluster_id 000002006BE04FC4
master_cluster_name ITSO_SVC1
master_vdisk_id 2
master_vdisk_name GM_App_Pri
aux_cluster_id 0000020061C06FCA
aux_cluster_name ITSO_SVC4
aux_vdisk_id 2
aux_vdisk_name GM_App_Sec
primary master
```

```
consistency_group_id
consistency_group_name
state consistent_synchronized
bg_copy_priority 50
progress
freeze_time
status online
sync
copy_type global
cycle_period_seconds 300
cycling_mode none
master_change_vdisk_id
master_change_vdisk_name
aux_change_vdisk_id
aux_change_vdisk_name
```

Restarting a Global Mirror Consistency Group in the Idling state

When restarting a Global Mirror Consistency Group in the Idling state, we must specify the copy direction.

If any updates have been performed on either the master or the auxiliary volume in any of the Global Mirror relationships in the Consistency Group, consistency will be compromised. Therefore, we must issue the **-force** parameter to start the relationship. If the **-force** parameter is not used, the command will fail.

In Example 4-40, we restart the Consistency Group and change the copy direction by specifying the auxiliary volumes to become the primaries.

Example 4-40 Restarting a Global Mirror relationship while changing the copy direction

```
IBM_2145:ITSO_SVC1:admin>startrcconsistgrp -primary aux CG_W2K3_GM
IBM_2145:ITSO_SVC1:admin>lsrcconsistgrp CG_W2K3_GM
id 0
name CG_W2K3_GM
master_cluster_id 000002006BE04FC4
master_cluster_name ITSO_SVC1
aux_cluster_id 0000020061C06FCA
aux_cluster_name ITSO_SVC4
primary aux
state consistent_synchronized
relationship_count 2
freeze_time
status
sync
copy_type global
cycle_period_seconds 300
cycling_mode none
RC_rel_id 0
RC_rel_name GMREL1
RC_rel_id 1
RC_rel_name GMREL2
```

4.2.8 Switching the direction on Global Mirror

In this section, we show how to change the copy direction of the stand-alone Global Mirror relationships and the Consistency Group.

Switching the direction on a Global Mirror volume relationship

When a Global Mirror relationship is in the Consistent synchronized state, we can change the copy direction for the relationship by using the **`switchrcrelationship`** command and specifying the primary volume.

If the volume that is specified as the primary when issuing this command is already a primary, the command has no effect.

In Example 4-41, we change the copy direction for the stand-alone Global Mirror relationship, specifying the auxiliary volume to become the primary.

> **Host write activity consideration:** Ensure there is no outstanding host write activity on the volumes that are about to transition from primary to secondary role.

Example 4-41 Switching the copy direction for a Global Mirror relationship

```
IBM_2145:ITSO_SVC1:admin>lsrcrelationship GMREL3
id 2
name GMREL3
master_cluster_id 000002006BE04FC4
master_cluster_name ITSO_SVC1
master_vdisk_id 2
master_vdisk_name GM_App_Pri
aux_cluster_id 0000020061C06FCA
aux_cluster_name ITSO_SVC4
aux_vdisk_id 2
aux_vdisk_name GM_App_Sec
primary master
consistency_group_id
consistency_group_name
state consistent_synchronized
bg_copy_priority 50
progress
freeze_time
status online
sync
copy_type global
cycle_period_seconds 300
cycling_mode none
master_change_vdisk_id
master_change_vdisk_name
aux_change_vdisk_id
aux_change_vdisk_name

IBM_2145:ITSO_SVC1:admin>switchrcrelationship -primary aux GMREL3
IBM_2145:ITSO_SVC1:admin>lsrcrelationship GMREL3
id 2
name GMREL3
master_cluster_id 000002006BE04FC4
master_cluster_name ITSO_SVC1
```

```
master_vdisk_id 2
master_vdisk_name GM_App_Pri
aux_cluster_id 0000020061C06FCA
aux_cluster_name ITSO_SVC4
aux_vdisk_id 2
aux_vdisk_name GM_App_Sec
primary aux
consistency_group_id
consistency_group_name
state consistent_synchronized
bg_copy_priority 50
progress
freeze_time
status online
sync
copy_type global
cycle_period_seconds 300
cycling_mode none
master_change_vdisk_id
master_change_vdisk_name
aux_change_vdisk_id
aux_change_vdisk_name
```

Switching the direction on a Global Mirror Consistency Group

When a Global Mirror Consistency Group is in the Consistent synchronized state, you can change the copy direction for the relationship by using the **switchrcconsistgrp** command and specifying the primary volume. If the volume that is specified as the primary when issuing this command is already a primary, the command has no effect.

In Example 4-42 on page 113, we change the copy direction for the Global Mirror Consistency Group, specifying the auxiliary to become the primary.

> **Host write activity consideration:** Ensure there is no outstanding host write activity on the volumes that are about to transition from primary to secondary role.

Example 4-42 Switching the copy direction for a Global Mirror Consistency Group

```
IBM_2145:ITSO_SVC1:admin>lsrcconsistgrp CG_W2K3_GM
id 0
name CG_W2K3_GM
master_cluster_id 000002006BE04FC4
master_cluster_name ITSO_SVC1
aux_cluster_id 0000020061C06FCA
aux_cluster_name ITSO_SVC4
primary master
state consistent_synchronized
relationship_count 2
freeze_time
status
sync
copy_type global
cycle_period_seconds 300
cycling_mode none
RC_rel_id 0
```

```
RC_rel_name GMREL1
RC_rel_id 1
RC_rel_name GMREL2
IBM_2145:ITSO_SVC1:admin>switchrcconsistgrp -primary aux CG_W2K3_GM
IBM_2145:ITSO_SVC1:admin>lsrcconsistgrp CG_W2K3_GM
id 0
name CG_W2K3_GM
master_cluster_id 000002006BE04FC4
master_cluster_name ITSO_SVC1
aux_cluster_id 0000020061C06FCA
aux_cluster_name ITSO_SVC4
primary aux
state consistent_synchronized
relationship_count 2
freeze_time
status
sync
copy_type global
cycle_period_seconds 300
cycling_mode none
RC_rel_id 0
RC_rel_name GMREL1
RC_rel_id 1
RC_rel_name GMREL2
```

4.3 Global Mirror with Change Volumes using the CLI

Starting with SVC 6.3, Global Mirror with Change Volumes attempts to create periodic (default 300 seconds) or one-off point-in-time-copies of the source volumes and replicate them to the secondary site. This has the effect of peak smoothing over the cycle period, and is a design enhancement that can reduce the required network bandwidth when compared to traditional Global Mirror.

Although the preferred name for this capability is Global Mirror with Change Volumes, it is enabled using the **cyclingmode** parameter on the **mkrcrelationship** or **chrcrelationship** commands.

4.3.1 Change Volume requirements

- You must have a Change Volume for both the primary and secondary volumes.
- Change Volumes would usually be thin-provisioned. Create the FlashCopy target volume using the **mkvdisk -autoexpand** command.
- Although they can be thin-provisioned, Change Volumes must be the same size (**mkvdisk -size**) as the primary and secondary volumes.
- Change Volumes must be in the same I/O Group as the volume of which they are a point-in-time copy.
- Change Volumes are not available for general use with other Metro or Global Mirror or FlashCopy mappings or for any other general access.

4.3.2 Changing Global Mirror to Global Mirror with Change Volumes

In this section, we show how to set **cyclingmode** on the stand-alone Global Mirror relationship (GMREL3) and the Consistency Group CG_W2K3_GM Global Mirror relationships (GMREL1 and GMREL2).

We assume that the source and target volumes have already been created and that the ISLs and zoning are in place, enabling the SVC systems to communicate. We also assume that the Global Mirror relationship has been already established.

To change the Global Mirror to Global Mirror with Change Volumes, perform the following steps:

1. Create thin-provisioned Change Volumes for the primary and secondary volumes at both sites.
2. Stop the stand-alone relationship GMREL3 to allow us to set **cyclingmode** at the primary site.
3. Set **cyclingmode** on the stand-alone relationship GMREL3 at the primary site.
4. Set the Change Volume on the master volume relationship GMREL3 at the primary site.
5. Set the Change Volume on the auxiliary volume relationship GMREL3 at the secondary site.
6. Start the stand-alone relationship GMREL3 with **cyclingmode** at the primary site.
7. Stop the Consistency Group CG_W2K3_GM to allow us to set **cyclingmode** at the primary site.
8. Set **cyclingmode** on the Consistency Group at the primary site.
9. Set the Change Volume on the master volume relationship GMREL1 of the Consistency Group CG_W2K3_GM at the primary site.
10. Set the Change Volume on the auxiliary volume relationship GMREL1 at the secondary site.
11. Set the Change Volume on the master volume relationship GMREL2 of the Consistency Group CG_W2K3_GM at the primary site.
12. Set the Change Volume on the auxiliary volume relationship GMREL2 at the secondary site.
13. Start the Consistency Group CG_W2K3_GM with **cyclingmode** at the primary site.

4.3.3 Creating the thin-provisioned Change Volumes

We start the setup by creating thin-provisioned Change Volumes for the primary and secondary volumes at both sites, as shown in Example 4-43.

Example 4-43 Creating the thin-provisioned volumes for Global Mirror with Change Volumes

```
IBM_2145:ITSO_SVC1:admin>mkvdisk -iogrp 0 -mdiskgrp 0 -size 10 -unit gb -rsize 20%
-autoexpand -grainsize 32 -name GM_DB_Pri_CHANGE_VOL
Virtual Disk, id [3], successfully created
IBM_2145:ITSO_SVC1:admin>mkvdisk -iogrp 0 -mdiskgrp 0 -size 10 -unit gb -rsize 20%
-autoexpand -grainsize 32 -name GM_DBLog_Pri_CHANGE_VOL
Virtual Disk, id [4], successfully created
IBM_2145:ITSO_SVC1:admin>mkvdisk -iogrp 0 -mdiskgrp 0 -size 10 -unit gb -rsize 20%
-autoexpand -grainsize 32 -name GM_App_Pri_CHANGE_VOL
Virtual Disk, id [5], successfully created

IBM_2145:ITSO_SVC4:admin>mkvdisk -iogrp 0 -mdiskgrp 0 -size 10 -unit gb -rsize 20%
-autoexpand -grainsize 32 -name GM_DB_Sec_CHANGE_VOL
Virtual Disk, id [3], successfully created
```

```
IBM_2145:ITSO_SVC4:admin>mkvdisk -iogrp 0 -mdiskgrp 0 -size 10 -unit gb -rsize 20%
-autoexpand -grainsize 32 -name GM_DBLog_Sec_CHANGE_VOL
Virtual Disk, id [4], successfully created
IBM_2145:ITSO_SVC4:admin>mkvdisk -iogrp 0 -mdiskgrp 0 -size 10 -unit gb -rsize 20%
-autoexpand -grainsize 32 -name GM_App_Sec_CHANGE_VOL
Virtual Disk, id [5], successfully created
```

4.3.4 Stopping the stand-alone remote copy relationship

We now display the remote copy relationships to ensure that they are in sync, and then we stop the stand-alone relationship GMREL3, as shown in Example 4-44.

Example 4-44 Stopping the remote copy stand-alone relationship

```
IBM_2145:ITSO_SVC1:admin>lsrcrelationship
id name   master_cluster_id master_cluster_name master_vdisk_id master_vdisk_name
aux_cluster_id   aux_cluster_name aux_vdisk_id aux_vdisk_name primary
consistency_group_id consistency_group_name state
bg_copy_priority progress copy_type cycling_mode
0  GMREL1 000002006BE04FC4  ITSO_SVC1          0              GM_DB_Pri
0000020061C06FCA ITSO_SVC4      0         GM_DB_Sec     aux    0
CG_W2K3_GM            consistent_synchronized 50                      global
none
1  GMREL2 000002006BE04FC4  ITSO_SVC1          1              GM_DBLog_Pri
0000020061C06FCA ITSO_SVC4      1         GM_DBLog_Sec   aux    0
CG_W2K3_GM            consistent_synchronized 50                      global
none
2  GMREL3 000002006BE04FC4  ITSO_SVC1          2              GM_App_Pri
0000020061C06FCA ITSO_SVC4     2       GM_App_Sec   aux
consistent_synchronized 50                        global    none

IBM_2145:ITSO_SVC1:admin>stoprcrelationship GMREL3
```

4.3.5 Setting cyclingmode on the stand-alone remote copy relationship

In Example 4-45, we set **cyclingmode** on the relationship using the **chrcrelationship** command. Note that the **cyclingmode** and **masterchange** parameters cannot be entered in the same command.

Example 4-45 Setting cyclingmode

```
IBM_2145:ITSO_SVC1:admin>chrcrelationship -cyclingmode multi  GMREL3
```

4.3.6 Setting the Change Volume on the master volume

In Example 4-46, we set the Change Volume for the primary volume. A display shows the name of the master Change Volume.

Example 4-46 Setting the Change Volume

```
IBM_2145:ITSO_SVC1:admin>chrcrelationship -masterchange GM_App_Pri_CHANGE_VOL

IBM_2145:ITSO_SVC1:admin>lsrcrelationship GMREL3
id 2
```

```
name GMREL3
master_cluster_id 000002006BE04FC4
master_cluster_name ITSO_SVC1
master_vdisk_id 2
master_vdisk_name GM_App_Pri
aux_cluster_id 0000020061C06FCA
aux_cluster_name ITSO_SVC4
aux_vdisk_id 2
aux_vdisk_name GM_App_Sec
primary aux
consistency_group_id
consistency_group_name
state consistent_stopped
bg_copy_priority 50
progress 100
freeze_time
status online
sync in_sync
copy_type global
cycle_period_seconds 300
cycling_mode multi
master_change_vdisk_id 5
master_change_vdisk_name GM_App_Pri_CHANGE_VOL
aux_change_vdisk_id
aux_change_vdisk_name
```

4.3.7 Setting the Change Volume on the auxiliary volume

In Example 4-47 on page 117, we set the Change Volume on the auxiliary volume in the secondary site. From the display, we can see the name of the volume.

Example 4-47 Setting the Change Volume on the auxiliary volume

```
IBM_2145:ITSO_SVC4:admin>chrcrelationship -auxchange GM_App_Sec_CHANGE_VOL 2
IBM_2145:ITSO_SVC4:admin>
IBM_2145:ITSO_SVC4:admin>lsrcrelationship GMREL3
id 2
name GMREL3
master_cluster_id 000002006BE04FC4
master_cluster_name ITSO_SVC1
master_vdisk_id 2
master_vdisk_name GM_App_Pri
aux_cluster_id 0000020061C06FCA
aux_cluster_name ITSO_SVC4
aux_vdisk_id 2
aux_vdisk_name GM_App_Sec
primary aux
consistency_group_id
consistency_group_name
state consistent_stopped
bg_copy_priority 50
progress 100
freeze_time
```

```
status online
sync in_sync
copy_type global
cycle_period_seconds 300
cycling_mode multi
master_change_vdisk_id 5
master_change_vdisk_name GM_App_Pri_CHANGE_VOL
aux_change_vdisk_id 5
aux_change_vdisk_name GM_App_Sec_CHANGE_VOL
```

4.3.8 Starting the stand-alone relationship with cyclingmode

In Example 4-48, we start the stand-alone relationship `GMREL3`. After a few minutes, we check the **`freeze_time`** parameter to see how it changes.

Example 4-48 Starting the stand-alone relationship in cyclingmode

```
IBM_2145:ITSO_SVC1:admin>startrcrelationship GMREL3

IBM_2145:ITSO_SVC1:admin>lsrcrelationship GMREL3
id 2
name GMREL3
master_cluster_id 000002006BE04FC4
master_cluster_name ITSO_SVC1
master_vdisk_id 2
master_vdisk_name GM_App_Pri
aux_cluster_id 0000020061C06FCA
aux_cluster_name ITSO_SVC4
aux_vdisk_id 2
aux_vdisk_name GM_App_Sec
primary aux
consistency_group_id
consistency_group_name
state consistent_copying
bg_copy_priority 50
progress 100
freeze_time 2011/10/04/20/37/20
status online
sync
copy_type global
cycle_period_seconds 300
cycling_mode multi
master_change_vdisk_id 5
master_change_vdisk_name GM_App_Pri_CHANGE_VOL
aux_change_vdisk_id 5
aux_change_vdisk_name GM_App_Sec_CHANGE_VOL

IBM_2145:ITSO_SVC1:admin>lsrcrelationship GMREL3
id 2
name GMREL3
master_cluster_id 000002006BE04FC4
master_cluster_name ITSO_SVC1
master_vdisk_id 2
master_vdisk_name GM_App_Pri
aux_cluster_id 0000020061C06FCA
```

```
aux_cluster_name ITSO_SVC4
aux_vdisk_id 2
aux_vdisk_name GM_App_Sec
primary aux
consistency_group_id
consistency_group_name
state consistent_copying
bg_copy_priority 50
progress 100
freeze_time 2011/10/04/20/42/25
status online
sync
copy_type global
cycle_period_seconds 300
cycling_mode multi
master_change_vdisk_id 5
master_change_vdisk_name GM_App_Pri_CHANGE_VOL
aux_change_vdisk_id 5
aux_change_vdisk_name GM_App_Sec_CHANGE_VOL
```

4.3.9 Stopping the Consistency Group to allow setting cyclingmode

In Example 4-49, we stop the Consistency Group with two relationships. The relationships must be stopped to allow us to set **cyclingmode**. A display shows that the state of the Consistency Group changes to consistent_stopped.

Example 4-49 Stopping the Consistency Group to set cyclingmode

```
IBM_2145:ITSO_SVC1:admin>stoprcconsistgrp CG_W2K3_GM

IBM_2145:ITSO_SVC1:admin>lsrcconsistgrp CG_W2K3_GM
id 0
name CG_W2K3_GM
master_cluster_id 000002006BE04FC4
master_cluster_name ITSO_SVC1
aux_cluster_id 0000020061C06FCA
aux_cluster_name ITSO_SVC4
primary aux
state consistent_stopped
relationship_count 2
freeze_time
status
sync in_sync
copy_type global
cycle_period_seconds 300
cycling_mode none
RC_rel_id 0
RC_rel_name GMREL1
RC_rel_id 1
RC_rel_name GMREL2
```

4.3.10 Setting cyclingmode on the Consistency Group

In Example 4-50, we set **cyclingmode** on the Consistency Group CG_W2K3_GM. To change it, we need to first stop the Consistency Group.

Example 4-50 Setting cyclingmode on the Consistency Group

```
IBM_2145:ITSO_SVC1:admin>chrcconsistgrp -cyclingmode multi CG_W2K3_GM

IBM_2145:ITSO_SVC1:admin>lsrcconsistgrp CG_W2K3_GM
id 0
name CG_W2K3_GM
master_cluster_id 000002006BE04FC4
master_cluster_name ITSO_SVC1
aux_cluster_id 0000020061C06FCA
aux_cluster_name ITSO_SVC4
primary aux
state consistent_stopped
relationship_count 2
freeze_time
status
sync in_sync
copy_type global
cycle_period_seconds 300
cycling_mode multi
RC_rel_id 0
RC_rel_name GMREL1
RC_rel_id 1
RC_rel_name GMREL2
```

4.3.11 Setting the Primary Change Volumes for the Consistency Group

In Example 4-51 on page 120 we change both of the relationships of the Consistency Group to add the Change Volumes on the primary volumes. A display shows the name of the master Change Volumes.

Example 4-51 Setting the Change Volume on the master volume

```
IBM_2145:ITSO_SVC1:admin>chrcrelationship -masterchange GM_DB_Pri_CHANGE_VOL
GMREL1

IBM_2145:ITSO_SVC1:admin>lsrcrelationship GMREL1
id 0
name GMREL1
master_cluster_id 000002006BE04FC4
master_cluster_name ITSO_SVC1
master_vdisk_id 0
master_vdisk_name GM_DB_Pri
aux_cluster_id 0000020061C06FCA
aux_cluster_name ITSO_SVC4
aux_vdisk_id 0
aux_vdisk_name GM_DB_Sec
primary aux
consistency_group_id 0
consistency_group_name CG_W2K3_GM
```

```
state consistent_stopped
bg_copy_priority 50
progress 100
freeze_time
status online
sync in_sync
copy_type global
cycle_period_seconds 300
cycling_mode multi
master_change_vdisk_id 3
master_change_vdisk_name GM_DB_Pri_CHANGE_VOL
aux_change_vdisk_id
aux_change_vdisk_name
IBM_2145:ITSO_SVC1:admin>

IBM_2145:ITSO_SVC1:admin>chrcrelationship -masterchange GM_DBLog_Pri_CHANGE_VOL
GMREL2

IBM_2145:ITSO_SVC1:admin>lsrcrelationship GMREL2
id 1
name GMREL2
master_cluster_id 000002006BE04FC4
master_cluster_name ITSO_SVC1
master_vdisk_id 1
master_vdisk_name GM_DBLog_Pri
aux_cluster_id 0000020061C06FCA
aux_cluster_name ITSO_SVC4
aux_vdisk_id 1
aux_vdisk_name GM_DBLog_Sec
primary aux
consistency_group_id 0
consistency_group_name CG_W2K3_GM
state consistent_stopped
bg_copy_priority 50
progress 100
freeze_time
status online
sync in_sync
copy_type global
cycle_period_seconds 300
cycling_mode multi
master_change_vdisk_id 4
master_change_vdisk_name GM_DBLog_Pri_CHANGE_VOL
aux_change_vdisk_id
aux_change_vdisk_name
```

4.3.12 Setting the secondary Change Volumes for the Consistency Group

In Example 4-52 we change both of the relationships of the Consistency Group to add the Change Volumes to the secondary volumes. The display shows the names of the auxiliary Change Volumes.

Example 4-52 Setting the Change Volumes on the auxiliary volumes

```
IBM_2145:ITSO_SVC4:admin>chrcrelationship -auxchange GM_DB_Sec_CHANGE_VOL GMREL1
```

```
IBM_2145:ITSO_SVC4:admin>lsrcrelationship GMREL1
id 0
name GMREL1
master_cluster_id 000002006BE04FC4
master_cluster_name ITSO_SVC1
master_vdisk_id 0
master_vdisk_name GM_DB_Pri
aux_cluster_id 0000020061C06FCA
aux_cluster_name ITSO_SVC4
aux_vdisk_id 0
aux_vdisk_name GM_DB_Sec
primary aux
consistency_group_id 0
consistency_group_name CG_W2K3_GM
state consistent_stopped
bg_copy_priority 50
progress 100
freeze_time
status online
sync in_sync
copy_type global
cycle_period_seconds 300
cycling_mode multi
master_change_vdisk_id 3
master_change_vdisk_name GM_DB_Pri_CHANGE_VOL
aux_change_vdisk_id 3
aux_change_vdisk_name GM_DB_Sec_CHANGE_VOL

IBM_2145:ITSO_SVC4:admin>chrcrelationship -auxchange GM_DBLog_Sec_CHANGE_VOL
GMREL2
IBM_2145:ITSO_SVC4:admin>lsrcrelationship GMREL2
id 1
name GMREL2
master_cluster_id 000002006BE04FC4
master_cluster_name ITSO_SVC1
master_vdisk_id 1
master_vdisk_name GM_DBLog_Pri
aux_cluster_id 0000020061C06FCA
aux_cluster_name ITSO_SVC4
aux_vdisk_id 1
aux_vdisk_name GM_DBLog_Sec
primary aux
consistency_group_id 0
consistency_group_name CG_W2K3_GM
state consistent_stopped
bg_copy_priority 50
progress 100
freeze_time
status online
sync in_sync
copy_type global
cycle_period_seconds 300
cycling_mode multi
master_change_vdisk_id 4
master_change_vdisk_name GM_DBLog_Pri_CHANGE_VOL
```

```
aux_change_vdisk_id 4
aux_change_vdisk_name GM_DBLog_Sec_CHANGE_VOL
```

4.3.13 Starting the Consistency Group with Change Volumes

In Example 4-53, we start the Consistency Group with Change Volumes. Remember that the **freeze_time** is the time stamp of the last primary change volume that has completed copying to the secondary site.

Example 4-53 Starting the Consistency Group with cycling mode

```
IBM_2145:ITSO_SVC1:admin>startrcconsistgrp CG_W2K3_GM
IBM_2145:ITSO_SVC1:admin>lsrcconsistgrp CG_W2K3_GM
id 0
name CG_W2K3_GM
master_cluster_id 000002006BE04FC4
master_cluster_name ITSO_SVC1
aux_cluster_id 0000020061C06FCA
aux_cluster_name ITSO_SVC4
primary aux
state consistent_copying
relationship_count 2
freeze_time 2011/10/04/21/02/33
status
sync
copy_type global
cycle_period_seconds 300
cycling_mode multi
RC_rel_id 0
RC_rel_name GMREL1
RC_rel_id 1
RC_rel_name GMREL2

IBM_2145:ITSO_SVC1:admin>lsrcconsistgrp CG_W2K3_GM
id 0
name CG_W2K3_GM
master_cluster_id 000002006BE04FC4
master_cluster_name ITSO_SVC1
aux_cluster_id 0000020061C06FCA
aux_cluster_name ITSO_SVC4
primary aux
state consistent_copying
relationship_count 2
freeze_time 2011/10/04/21/07/42
status
sync
copy_type global
cycle_period_seconds 300
cycling_mode multi
RC_rel_id 0
RC_rel_name GMREL1
RC_rel_id 1
RC_rel_name GMREL2
```

4.4 Implementation using the GUI

It is often easier to manage Metro Mirror or Global Mirror with the GUI, if you have a small number of mappings. When using many mappings it can be easier to use the CLI to execute your commands. For a large number of mappings we suggest you consider adding TPC for Replication to give you additional automation options.

For more details about TPC for Replication see Chapter 11, “Software solutions and services based on IBM SVC Replication Family Services” on page 427.

There are two key panels that you can use to manage your remote copies:

1. The Remote Copy panel, as shown in Figure 4-7 on page 124.

   Metro Mirror and Global Mirror allow you to create a relationship between two volumes, so that updates that are made by an application to one volume are mirrored on the other volume. The volumes can be in the same system (cluster) or on two separate systems.

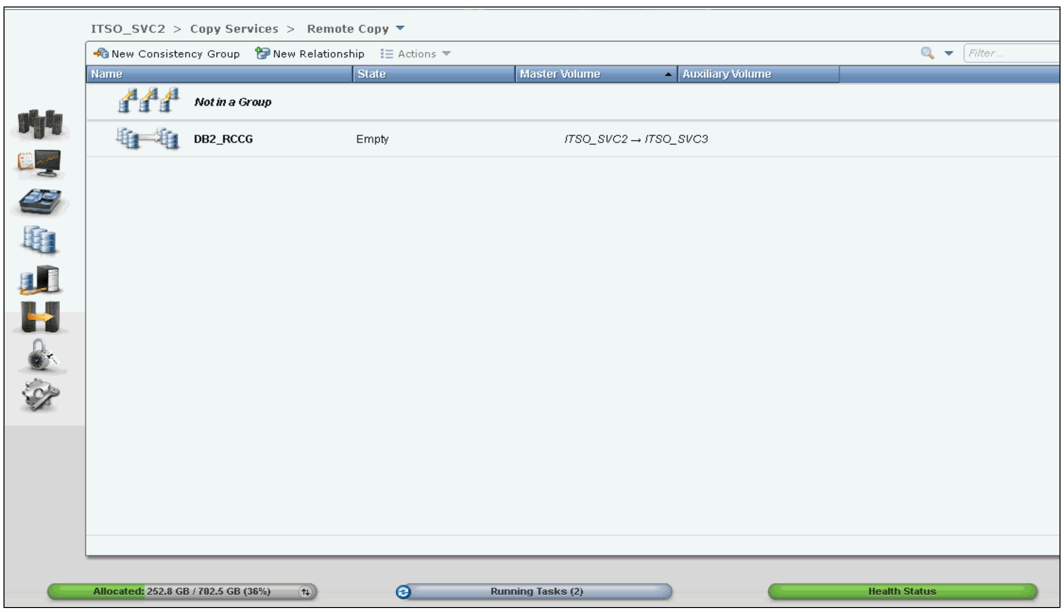


Figure 4-7 Remote Copy panel

2. The Partnerships panel, as shown in Figure 4-8

   Partnerships define an association between a local system (cluster) and a remote system. Partnerships can be used to create a disaster recovery environment, or to migrate data between systems that are in separate locations.

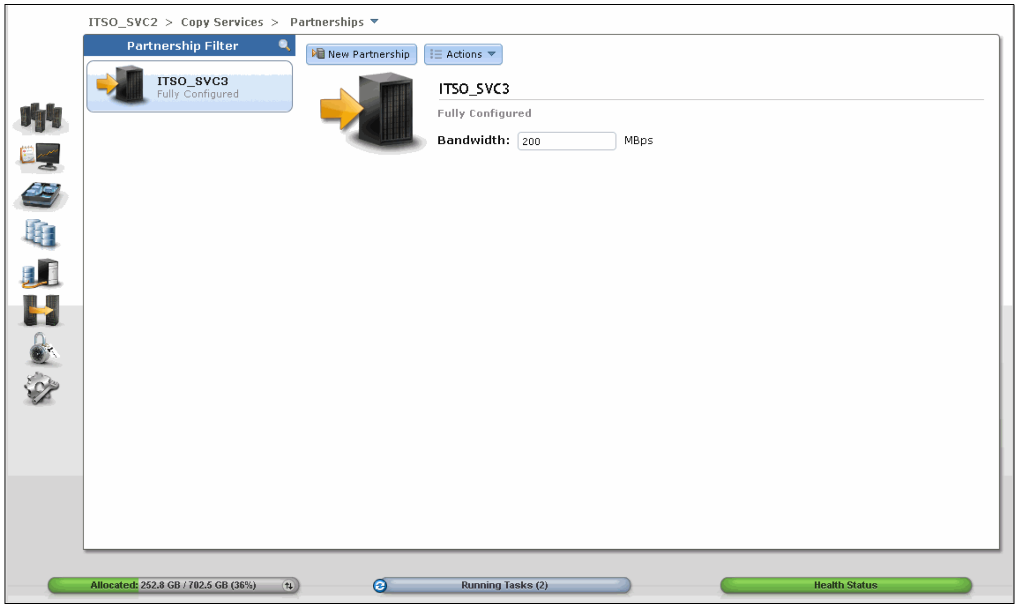


Figure 4-8 Partnerships panel

4.4.1 Creating a partnership between two systems (clusters)

> **Intra-cluster consideration:** If you are creating an intra-cluster Metro Mirror, do *not* perform this next step to create the Metro Mirror partnership.
>
> Instead, go to 4.4.2, “Creating stand-alone relationships between volumes” on page 128.

To create a partnership, follow these steps:

1. From the Overview panel, click **Copy Services** → **Partnerships**. The Partnerships panel opens, as shown in Figure 4-9.

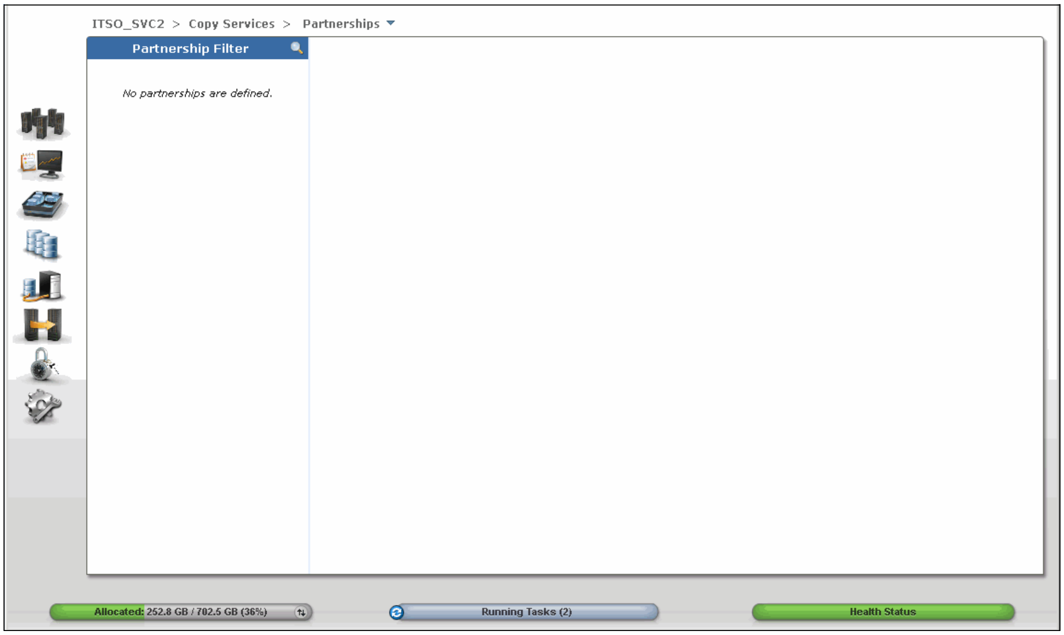


Figure 4-9 Partnerships panel

2. Click **New Partnership** to create a new partnership with another system (cluster) as shown in Figure 4-10.

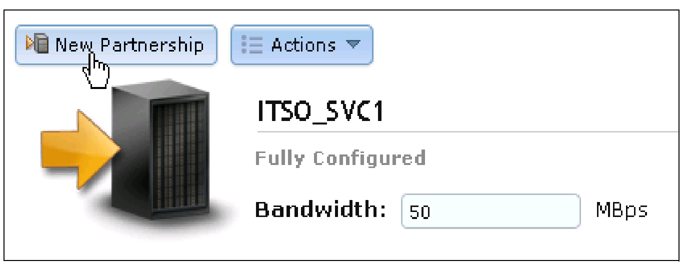


Figure 4-10 New partnership

3. On the Create Partnership window (Figure 4-11 on page 127), complete the following elements:

   – Select an available system in the drop-down list box. If there is no candidate, you will receive the following error message: `This cluster does not have any candidates.`

   – Enter a bandwidth (MBps) that is used by the background copy process between the clusters in the partnership. Set this value significantly lower than the bandwidth that can be sustained by the network. The network must be able to sustain host requests, the partnership heartbeat, and the rate of background copy.

   Setting the background copy rate: If your network resources are limited, it is best not to set the background copy rate too high because it needs to share the available bandwidth with other active relationships.

   This caution does not apply to Global Mirror with Change Volumes.

– Click **Create** to confirm the partnership.

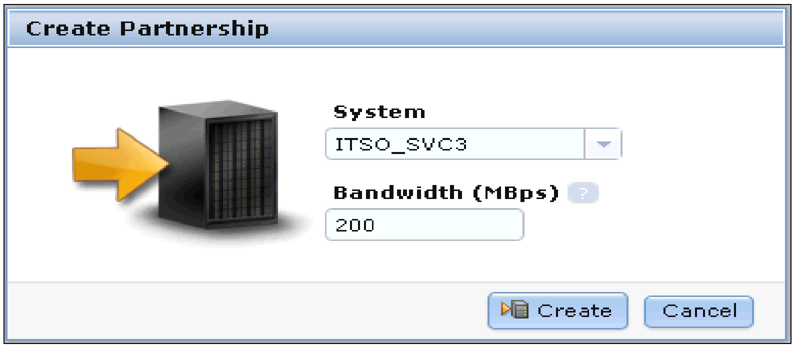


Figure 4-11 Create Partnership window

4. As shown in Figure 4-12, our partnership is in the `Partially Configured` state, because we have only performed the work on one side of the partnership at this point.

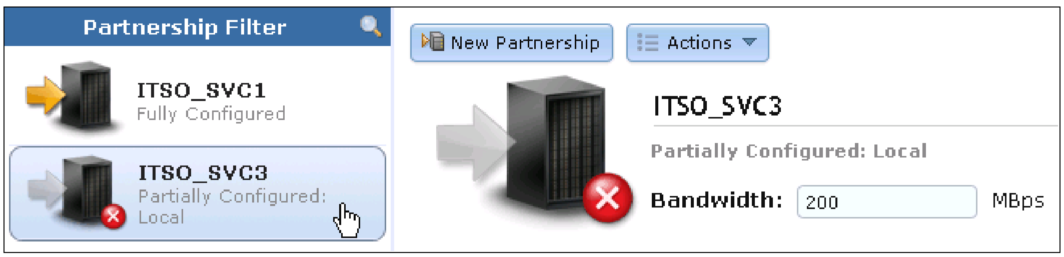


Figure 4-12 Viewing cluster partnerships

To fully configure the partnership, we must perform the same steps on the other system in the partnership. For simplicity and brevity, only the two most significant windows are shown when the partnership is fully configured.

5. Launching the GUI for ITSO_SVC3, we select **ITSO_SVC2** for the partnership. We specify the available bandwidth for the background copy, again **200** MBps (megabytes per second) and then click **Create**.

   Now that both sides of the partnership are defined, the resulting windows, which are shown in Figure 4-13 on page 127 and Figure 4-14 on page 128, confirm that our partnership is now in the Fully Configured state. Figure 4-13 on page 127 shows the view from system ITSO_SVC2.

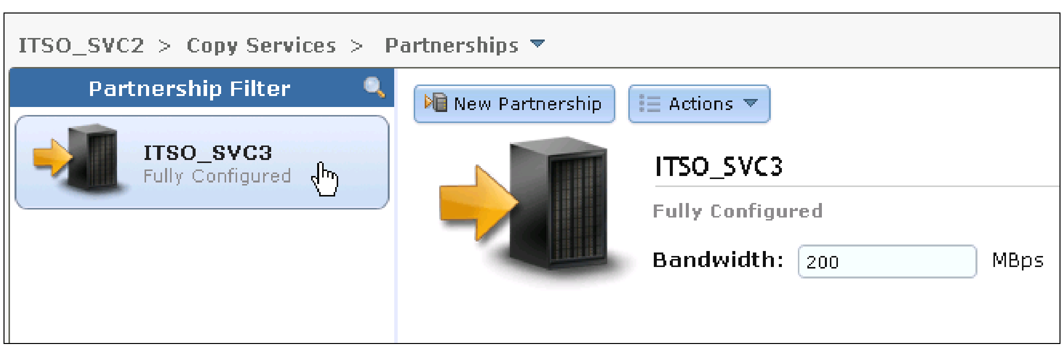


Figure 4-13 View from ITSO_SVC2: Fully configured partnership

Figure 4-14 shows the view from system ITSO_SVC3.

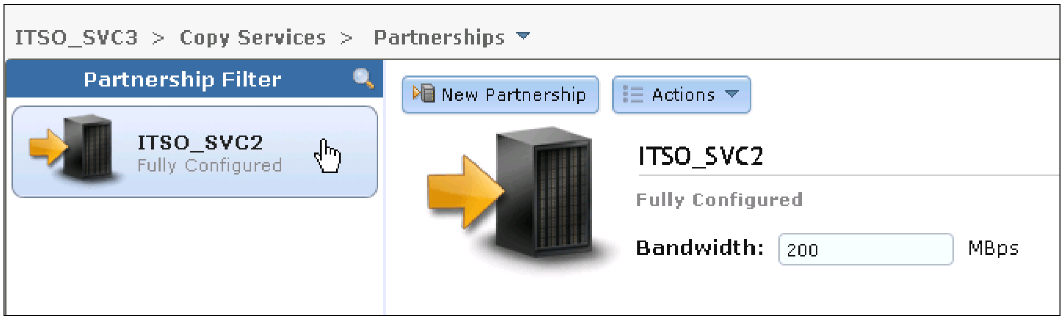


Figure 4-14 View from ITSO_SVC3: Fully configured partnership

4.4.2 Creating stand-alone relationships between volumes

In this section, we create remote copy mappings for volumes with their respective remote targets. The source and target volumes have been created prior to this operation on both clusters.

To perform this action, follow these steps:

1. From the Overview panel, click **Copy Services** → **Remote Copy**.
2. Click **New Relationship**, as shown in Figure 4-15.

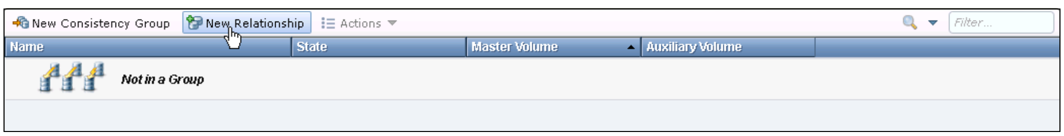


Figure 4-15 New relationship action

3. In the New Relationship window, select the type of relationship that you want to create (Figure 4-16 on page 129) then click **Next**:
   - Metro Mirror

     Metro Mirror creates a synchronous copy of data from a primary volume to a secondary volume. A secondary volume can either be located on the same cluster or on another cluster.
   - Global Mirror

     Global Mirror provides a write-order-consistent copy of a source volume on a target volume. Data is written to the target volume asynchronously, so that the copy is continuously updated, but the copy might not contain the last few updates in the event that a disaster recovery operation is performed. The target volume will typically lag the primary volume by less than 1 second.
   - Global Mirror with Change Volumes

     Global Mirror with Change Volumes provides a point-in-time copy of a source volume on a target volume. A FlashCopy is automatically taken every cycle period so that the target volume is repeatedly updated. A FlashCopy is regularly updated from the target volume also so that there is always a static point-in-time copy of the data available. The FlashCopy mappings used by Global Mirror with Change Volumes are not available for manipulation by external processes.

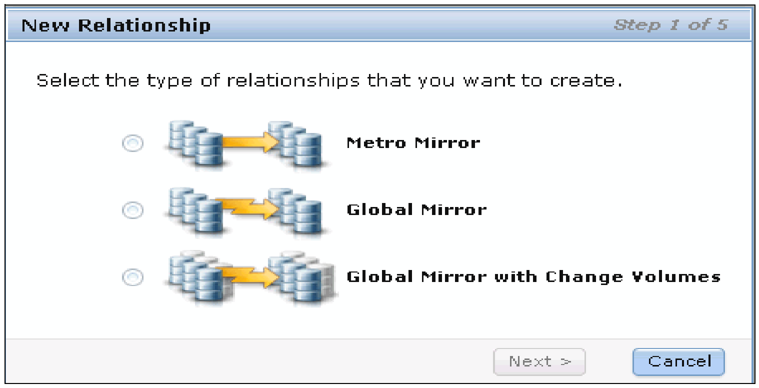


Figure 4-16 Select the type of relationship that you want to create

4. In the next window, select the location of the auxiliary volumes, as shown in Figure 4-17:
   - On this system, which means that the volumes are local.
   - On another system, which means that you select the remote system from the drop-down list.
   - After you make a selection, click **Next**.

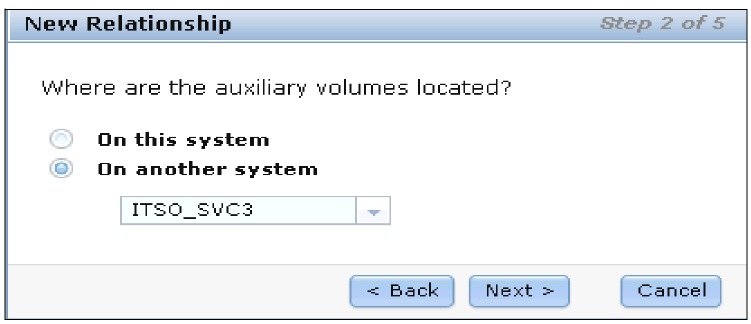


Figure 4-17 Specifying the location of the auxiliary volumes

5. In the New Relationship window that is shown in Figure 4-18 on page 130, you can create new relationships. Select a master volume in the Master drop-down list, and then select an auxiliary volume in the Auxiliary drop-down list for this master and click **Add**. If needed, repeat this action to create other relationships.

Volume size consideration: The master and auxiliary volumes must be of equal size (thin provisioning aside). So for a given source volume, only the targets with the appropriate size are shown in the list box.

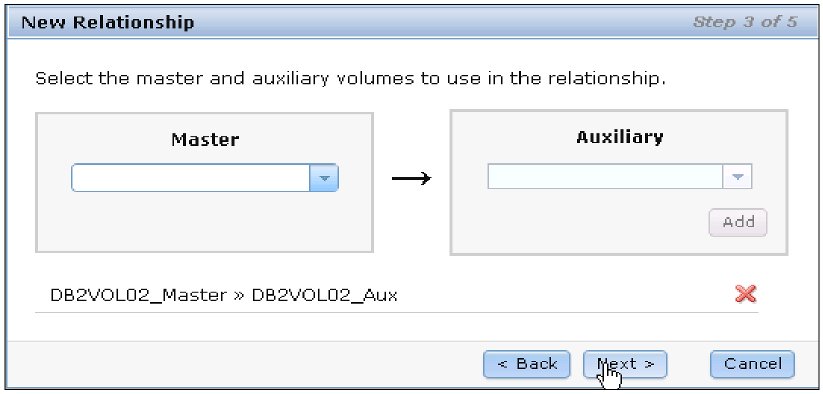


Figure 4-18 Create the relationships between the master and auxiliary volumes

To remove a relationship that has been created, click , as shown in Figure 4-18. After all the relationships that you want to create are shown, click **Next**.

6. Select whether the volumes are already synchronized, as shown in Figure 4-19, and then click **Next**. In most cases you will select **No** unless you are planning to populate remote volumes using tape copies. See 3.6.2, “Tape-based synchronization” on page 48, for more information about this topic.

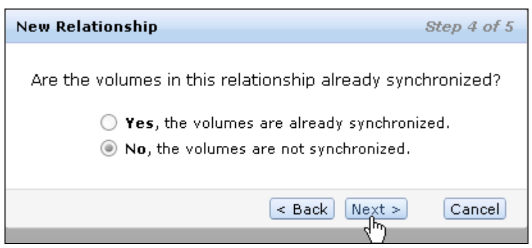


Figure 4-19 Are the volumes already synchronized?

7. Finally, on the last window, select whether you want to start to copy the data, as shown in Figure 4-20. Then click **Finish**.

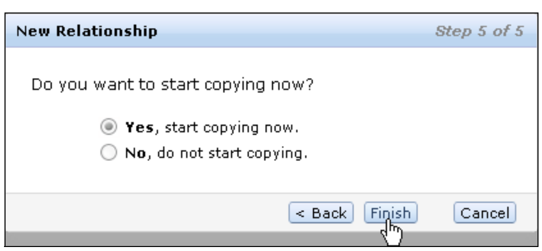


Figure 4-20 Start copying now?

The relationships are visible in the Remote Copy panel. If you selected to copy the data, you can see that their status is `Inconsistent Copying`. You can check the copying progress in the Running Tasks status area, as shown in Figure 4-21 on page 131.

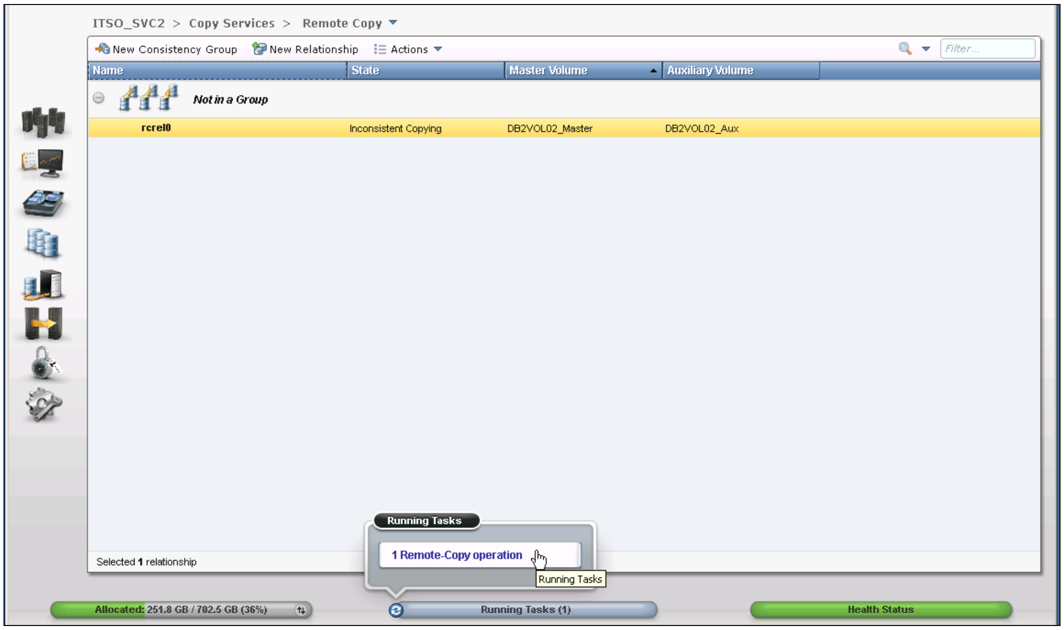


Figure 4-21 Remote Copy panel with an inconsistent copying status

After the copy is finished, the relationships status changes to `Consistent synchronized`.

4.4.3 Creating a Consistency Group

To create a Consistency Group, follow these steps:

1. From the Overview panel, click **Copy Services** → **Remote Copy**.

2. Click **New Consistency Group** (Figure 4-22).

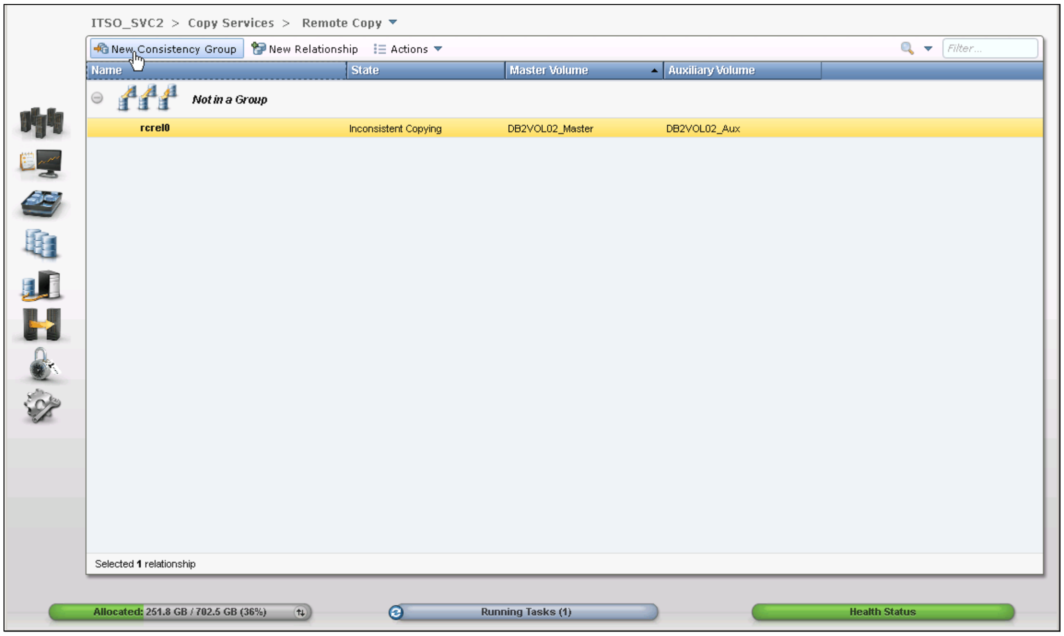


Figure 4-22 New Consistency Group

3. Enter a name for the Consistency Group, and then click **Next** (Figure 4-23 on page 132).

> **Naming consideration:** Use a well-considered set of naming standards. If you do not provide a name, the system automatically generates the name `rccstgrp` with an ID sequence number suffix.
>
> Starting with an alphabetic character, you can use the letters A to Z and a to z, the numbers 0 to 9, and the underscore (_) character. Systems running code level 6 support object names up to 63 characters but when they are partnered with systems running code level 5.1 or lower, those lower version partners will display object names truncated at 15 characters.

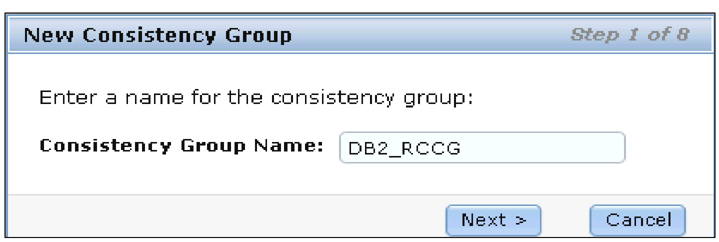


Figure 4-23 Enter a Consistency Group name

4. In the next window, select where the auxiliary volumes are located, as shown in Figure 4-24:
   - On this system, which means that the volumes are local
   - On another system, which means that you select the remote system in the drop-down list

After you make a selection, click **Next**.

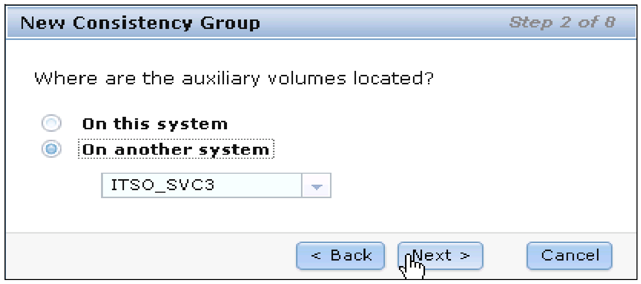


Figure 4-24 Auxiliary volumes location

5. Select whether you want to add relationships to this group, as shown in Figure 4-25 on page 133. There are two options:
   - If you answer Yes, click **Next** to continue the wizard, and go to step 6.
   - If you answer No, click **Finish** to create an empty Consistency Group that can be used later.

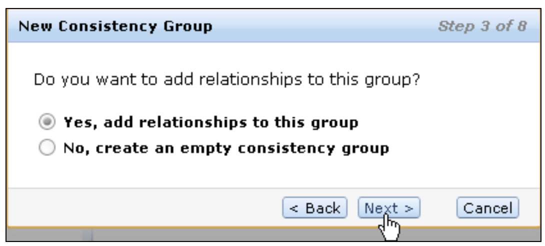


Figure 4-25 Add relationships to this group

6. Select the type of relationship that you want to create (Figure 4-26) and click **Next**.
   - Metro Mirror

     Metro Mirror creates a synchronous copy of data from a primary volume to a secondary volume. A secondary volume can either be located on the same cluster or on another cluster.
   - Global Mirror

     Global Mirror provides a write-order-consistent copy of a source volume on a target volume. Data is written to the target volume asynchronously so that the copy is continuously updated. However, the copy might not contain the last few updates in the event that a disaster recovery operation is performed. The target volume will typically lag the primary volume by less than 1 second.
   - Global Mirror with Change Volumes

     Global Mirror with Change Volumes provides a point-in-time copy of a source volume on a target volume. A FlashCopy is automatically taken every cycle period so that the target volume is repeatedly updated. A FlashCopy is regularly updated from the target volume also so that there is always a static point-in-time copy of the data available. The FlashCopy mappings used by Global Mirror with Change Volumes are not available for manipulation by external processes.

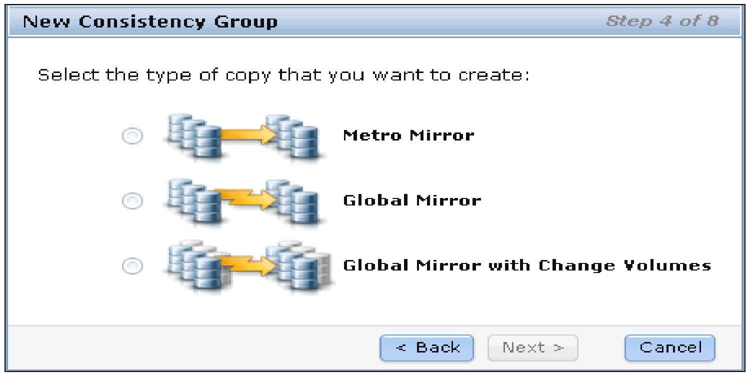


Figure 4-26 Select the type of relationship that you want to create

7. As shown in Figure 4-27 on page 134, you can optionally select existing relationships to add to the group, and then click **Next**.

Tip: To select multiple relationships, hold down the **Ctrl** key and use your mouse to select the entries that you want to include.

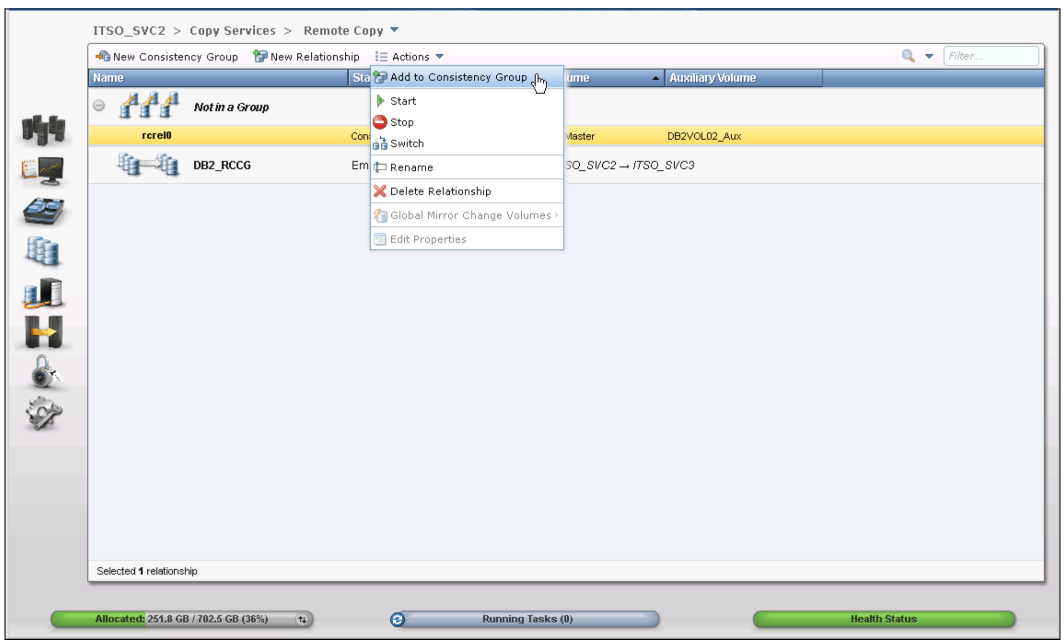


Figure 4-27 Select existing relationships to add to the group

8. In the window that is shown in Figure 4-28, you can create new relationships. Select a volume in the Master drop-down list box, and then select a volume in the Auxiliary drop-down list box for this master. Click **Add**, as shown in Figure 4-28. Repeat this action to create other relationships, if needed.

Volume size consideration: The master and auxiliary volumes must be of equal size (thin-provisioning aside). So for a given source volume, only the targets with the appropriate size are displayed.

To remove a relationship that has been created, click , as shown in Figure 4-28. After all the relationships that you want to create are displayed, click **Next**.

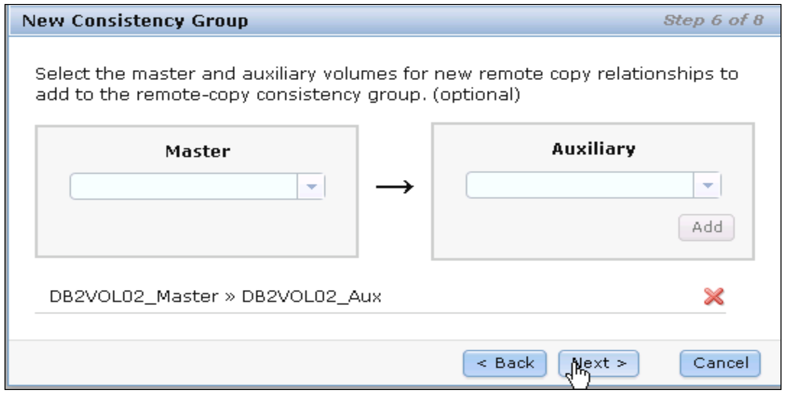


Figure 4-28 Create relationships between the master and auxiliary volumes

9. Select if the volumes are already synchronized, as shown in Figure 4-29 on page 135, and then click **Next**. In most cases you will select **No** unless you are planning to populate remote volumes using tape copies. See Chapter 3, “Metro Mirror and Global Mirror” on page 35, for more information about that topic.

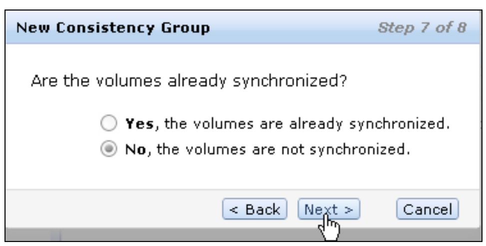


Figure 4-29 Are the volumes already synchronized?

10. Finally, on the last window, select whether you want to start to copy the data, as shown in Figure 4-30, and then click **Finish**.

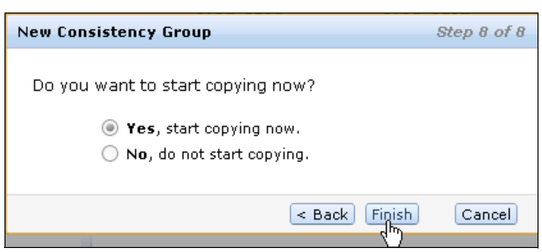


Figure 4-30 Start copying now?

11. The relationships are visible in the Remote copy panel. If you selected to copy the data, you can see that the status of the relationships is `Inconsistent Copying`. You can check the copying progress in the Running Tasks status area, as shown in Figure 4-31.

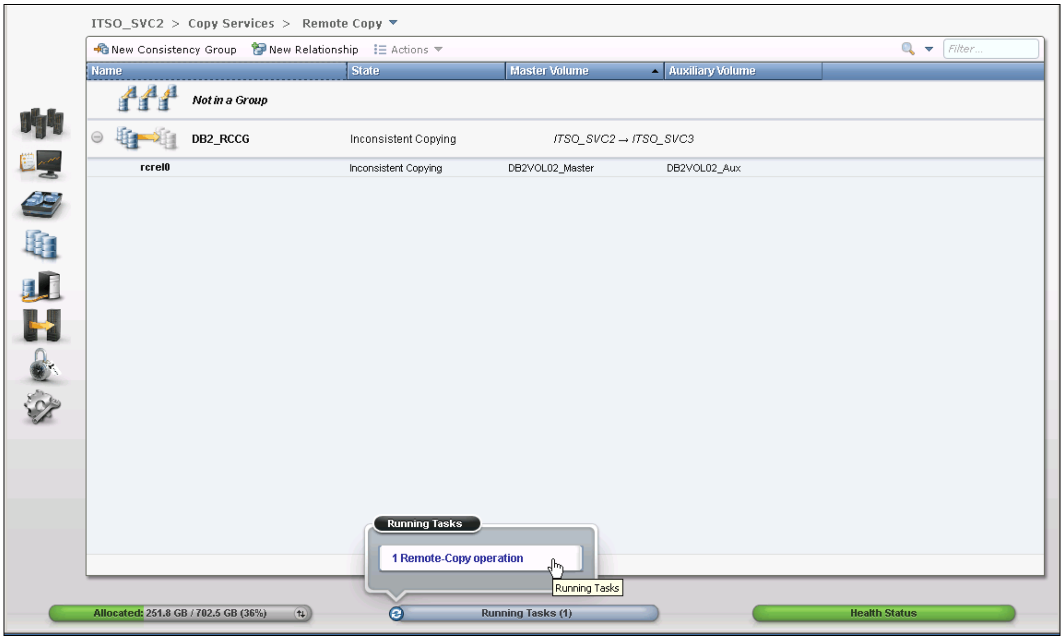


Figure 4-31 Consistency Group created with relationship in copying status

After the copies are completed, the relationships and the Consistency Group change to the `Consistent synchronized` status.

4.4.4 Renaming a Consistency Group

To rename a Consistency Group, perform the following steps:

1. From the Overview panel, click **Copy Services menu** → **Remote Copy**.
2. Select the Consistency Group that you want to rename in the panel. Then select **Actions** → **Rename**, as shown in Figure 4-32.

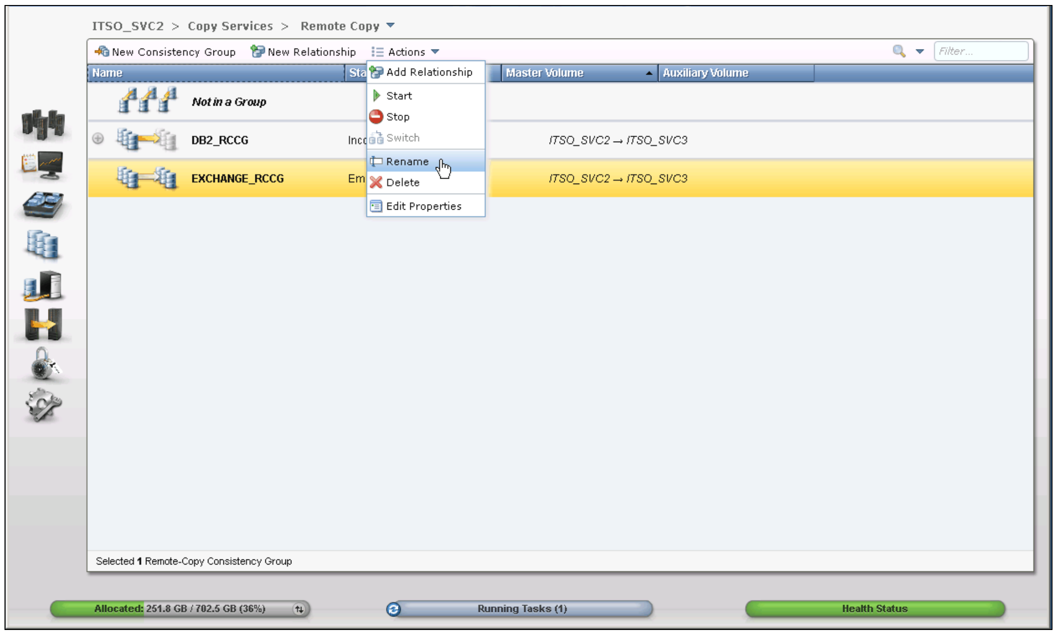


Figure 4-32 Renaming a Consistency Group

3. Type the new name that you want to assign to the Consistency Group and click **Rename** (Figure 4-33).

Figure 4-33 Type the new name and click Rename

4. From the Remote Copy panel, the new Consistency Group name is displayed.

4.4.5 Renaming a relationship

Perform the following steps to rename a Metro Mirror or Global Mirror relationship:

1. From the Overview panel, click **Copy Services menu** → **Remote Copy**.
2. In the table, select the remote copy relationship mapping that you want to rename. Click **Actions** → **Rename** (Figure 4-34 on page 138).

> **Tip:** You can also right-click a remote copy relationship and select `Rename` from the list.

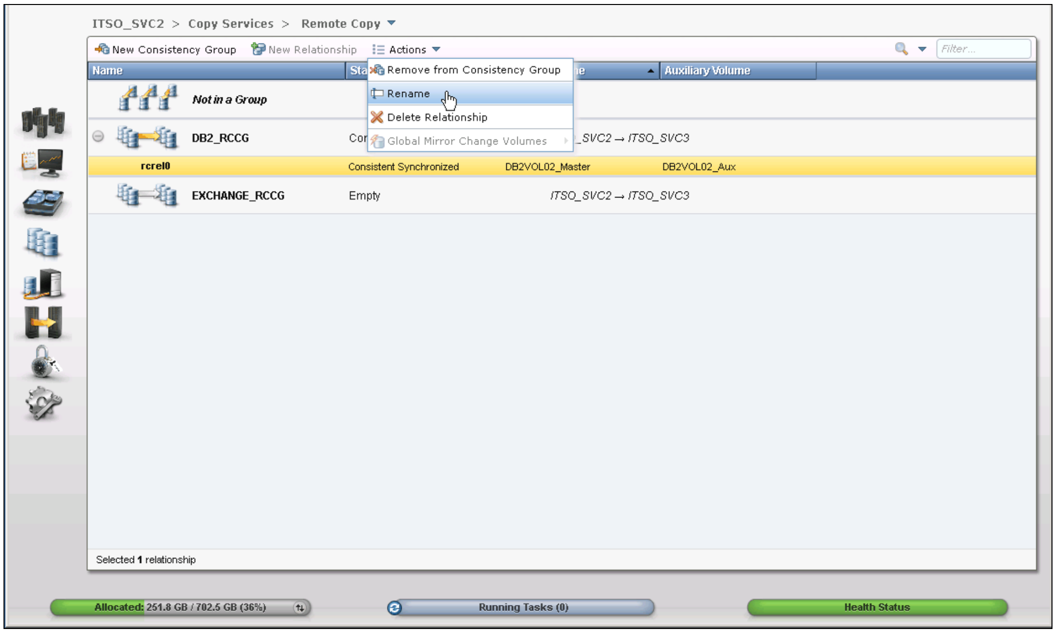


Figure 4-34 Rename a relationship

3. In the Rename Relationship window, type the new name that you want to assign to the FlashCopy mapping and click **Rename** (Figure 4-35).

Figure 4-35 Type the new name and click Rename

4.4.6 Moving a stand-alone remote copy relationship to a Consistency Group

Perform the following steps to move a remote copy relationship to a Consistency Group:

1. From the Overview panel, click **Copy Services** → **Remote Copy**.
2. Expand the Column **Not in a Group**.
3. Select the relationship that you want to move to the Consistency Group**.**

4. Click **Actions** → **Add to Consistency Group**, as shown in Figure 4-36 on page 139.

Tip: You can also right-click a remote copy relationship and select **Add to Consistency Group** from the list.

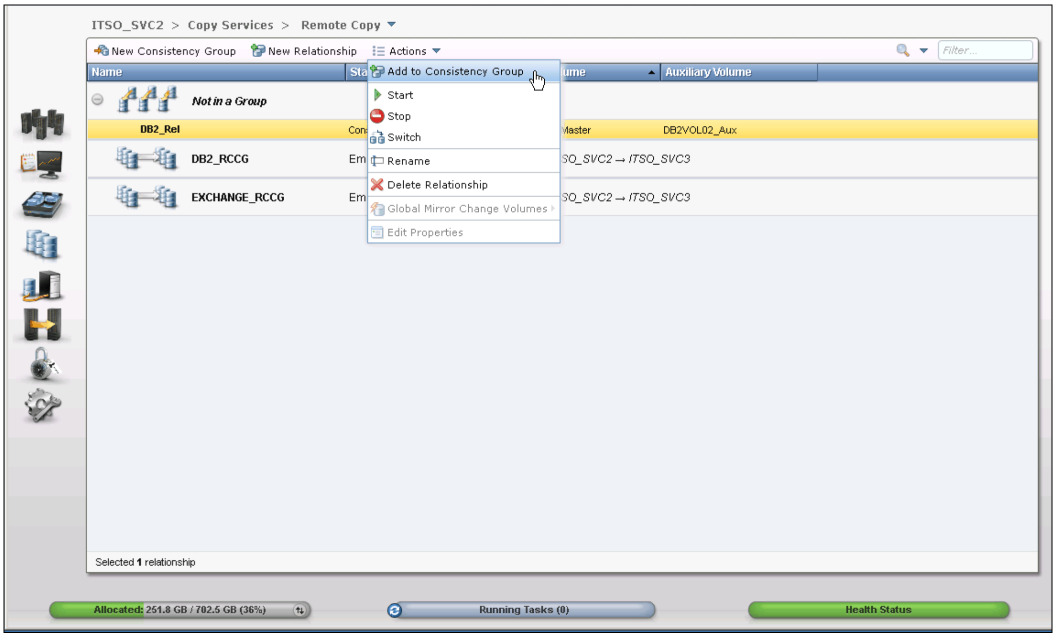


Figure 4-36 Adding a volume to a Consistency Group

5. In the Add Relationship to Consistency Group window, select the Consistency Group for this remote copy relationship using the drop-down list box (Figure 4-37). Click **Add to Consistency Group** to confirm your changes.

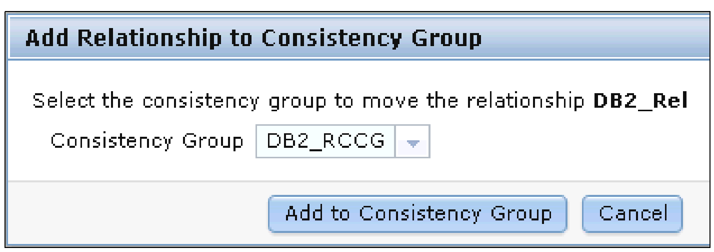


Figure 4-37 Select a Consistency Group

4.4.7 Removing a remote copy relationship from a Consistency Group

Perform the following steps to remove a remote copy relationship from a Consistency Group:

1. From the Overview panel, click **Copy Services** → **Remote Copy**.
2. Select a Consistency Group**.**
3. Select the remote copy relationship that you want to remove from the Consistency Group.

4. Click **Actions** → **Remove from Consistency Group** (Figure 4-38 on page 140).

Tip: You can also right-click a remote copy relationship and select **Remove from Consistency Group** from the list.

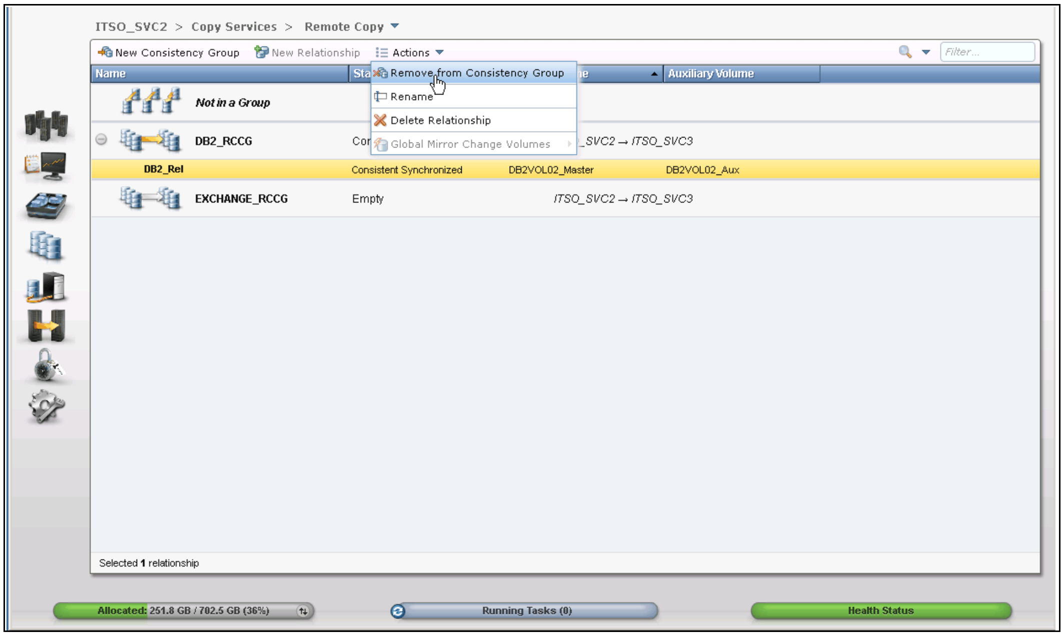


Figure 4-38 Removing a volume from a Consistency Group

5. In the Remove Relationship From Consistency Group window, click **Remove** (Figure 4-39).

Figure 4-39 Confirm removal of the volume from the Consistency Group

4.4.8 Starting a relationship

When a remote copy relationship is created, the remote copy process can be started. Only relationships that are not members of a Consistency Group, or relationships that are the sole members of a Consistency Group, can be started individually.

Perform the following steps to start a Metro or Global Mirror relationship:

1. From the Overview panel, click **Copy Services** → **Remote Copy**.
2. Expand the Column **Not in a Group**.
3. In the table, select the remote copy relationship that you want to start.

4. Click **Actions** → **Start** (Figure 4-40 on page 141) to start the remote copy process.

Tip: You can also right-click a relationship and select **Start** from the list.

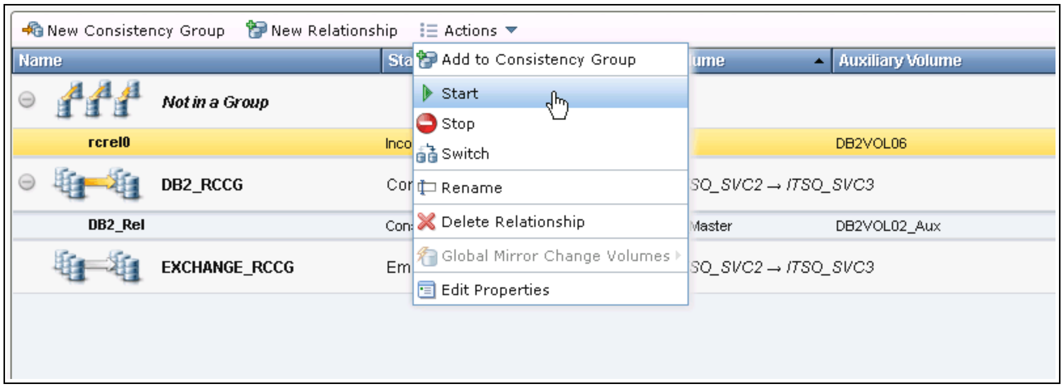


Figure 4-40 Starting a relationship

5. If the relationship was not consistent, you can check the remote copy progress in the Running Tasks status area (Figure 4-41).

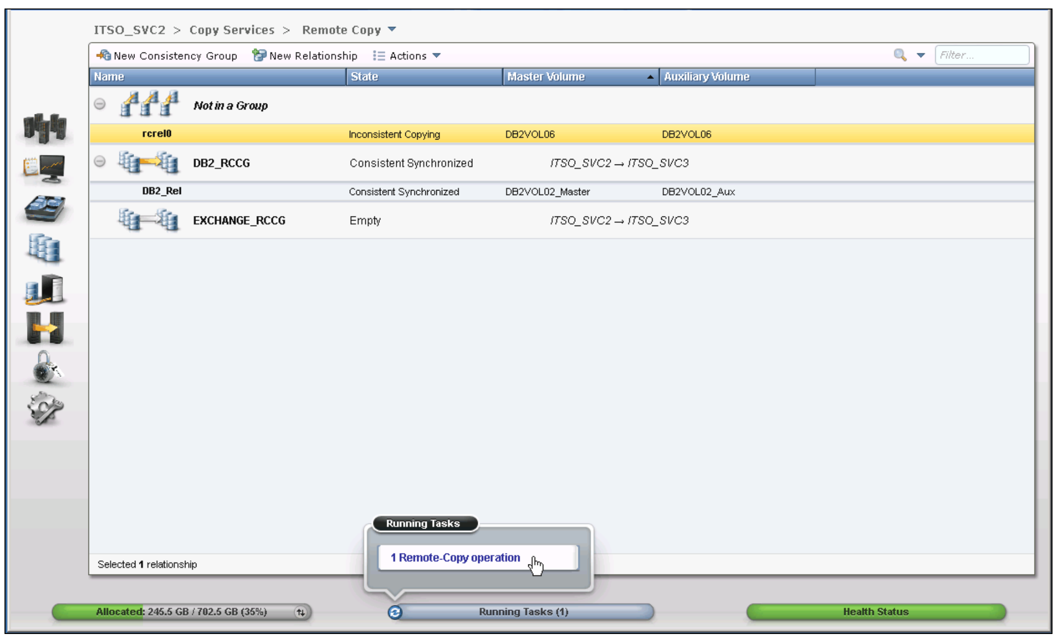


Figure 4-41 Checking relationship synchronization progress

6. After the task is completed, the remote copy relationship status has a `Consistent Synchronized` state (Figure 4-42).

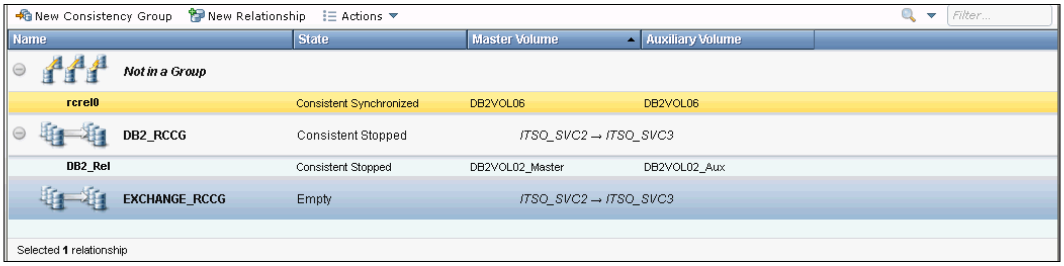


Figure 4-42 Consistent Synchronized stand-alone relationship

4.4.9 Starting a Consistency Group

All mappings in a Consistency Group will be brought to the same state. To start the remote copy Consistency Group, follow these steps:

1. From the Overview panel, click **Copy Services** → **Remote Copy**.
2. Select the Consistency Group that you want to start (Figure 4-43).

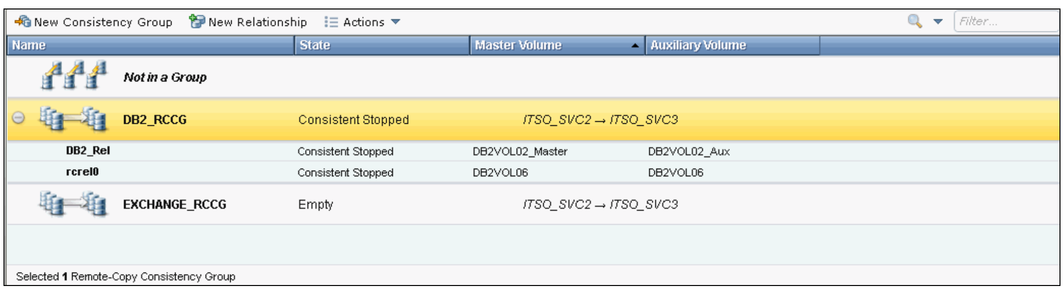


Figure 4-43 Consistency Groups view

3. Click **Actions** → **Start** (Figure 4-44) to start the remote copy Consistency Group.

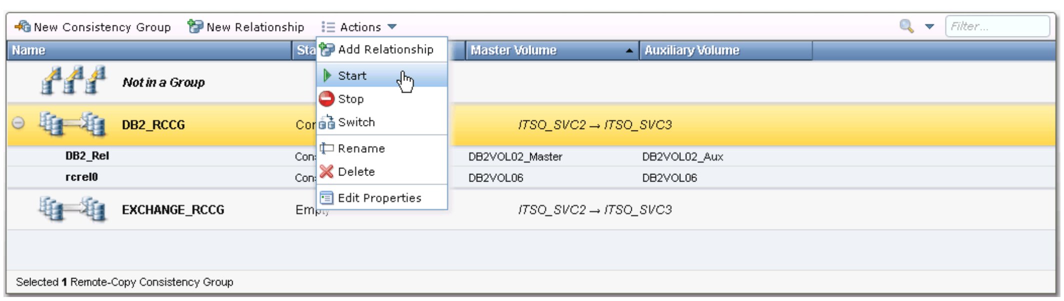


Figure 4-44 Start a Consistency Group

4. You can check the remote copy Consistency Group progress, as shown in Figure 4-45.

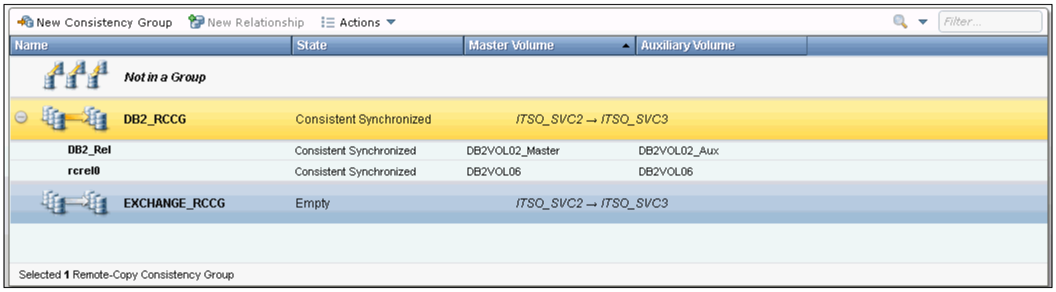


Figure 4-45 Checking Consistency Group synchronization progress

5. After the task is completed, the Consistency Group and all its relationship statuses are in a `Consistent Synchronized` state (Figure 4-46 on page 143).

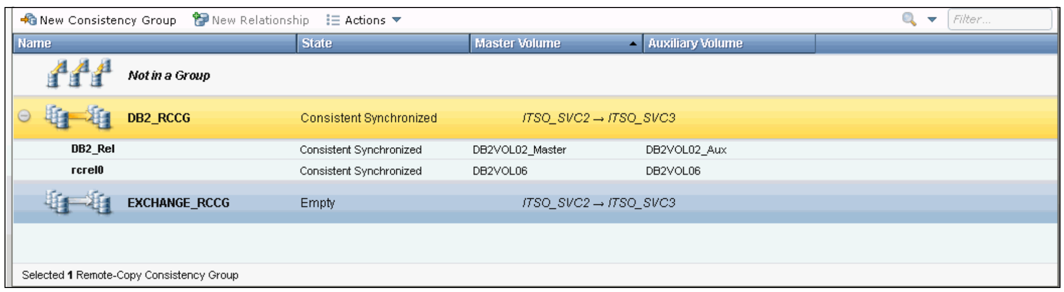


Figure 4-46 Consistent Synchronized Consistency Group

4.4.10 Switching the copy direction on a relationship

When a relationship is in the Consistent Synchronized state, the copy direction can be changed. Only relationships that are not members of a Consistency Group, or relationships that are the sole members of a Consistency Group, can be switched individually.

> **Host write activity consideration:** Ensure there is no outstanding host write activity on the volumes that are about to transition from primary to secondary role.

Perform the following steps to switch a remote copy relationship:

1. From the Overview panel, click **Copy Services** → **Remote Copy**.
2. Expand the **Not in a Group** column.
3. In the table, select the remote copy relationship that you want to switch.

4. Click **Actions** → **Switch** (Figure 4-47) to start the remote copy process.

Tip: You can also right-click a relationship and select **Switch** from the list.

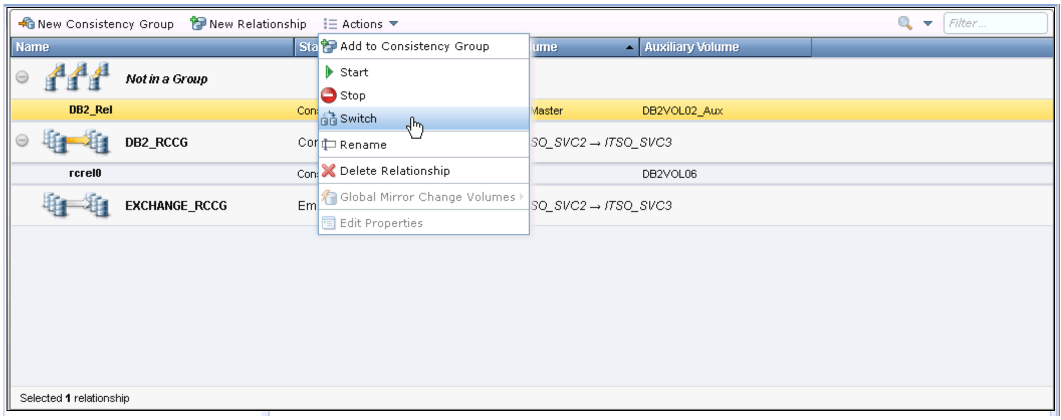


Figure 4-47 Switching the direction of a relationship

5. A Warning window opens (Figure 4-48 on page 144). A confirmation is needed to switch the remote copy relationship direction. As shown in Figure 4-48 on page 144, the remote copy is switched from the master volume to the auxiliary volume. Click **OK** to confirm your choice.

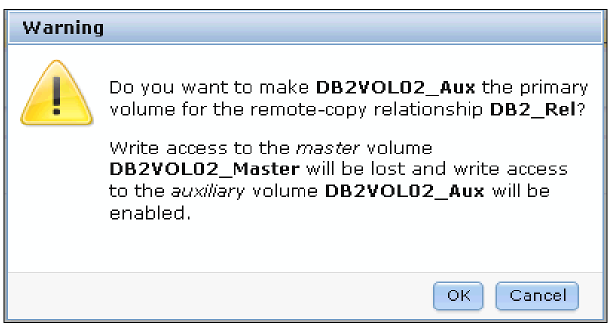


Figure 4-48 Warning window when switching the direction of a relationship

6. The copy direction is now switched, as shown in Figure 4-49. The auxiliary volume is now accessible and indicated as the primary volume. The auxiliary volume is now synchronized through to the master volume.

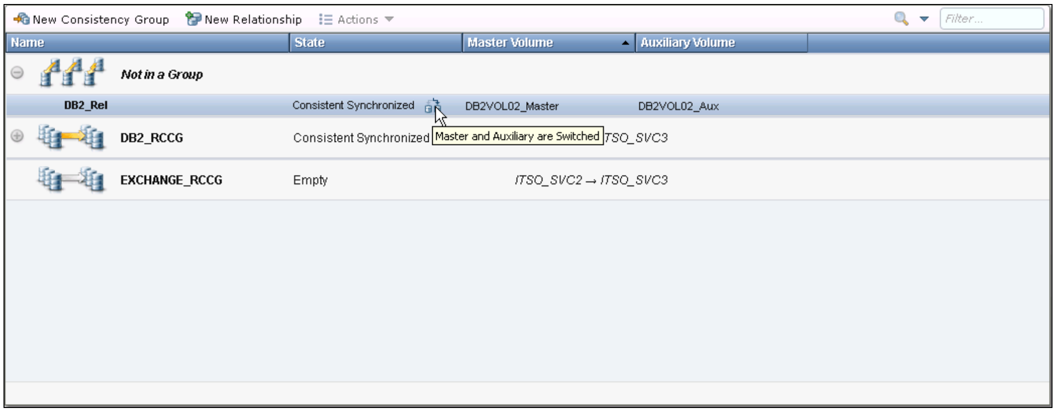


Figure 4-49 Master volume and auxiliary volume are switched

4.4.11 Switching the copy direction on a Consistency Group

When a Consistency Group is in the Consistent Synchronized state, the copy direction for this Consistency Group can be changed.

Host write activity consideration: Ensure there is no outstanding host write activity on the volumes that are about to transition from primary to secondary role.

Perform the following steps to switch a Consistency Group:

1. From the Overview panel, click **Copy Services** → **Remote Copy**.
2. Select the Consistency Group that you want to switch.
3. Click **Actions** → **Switch** (Figure 4-50 on page 145) to switch the copy direction of a consistency group.

Tip: You can also right-click a relationship and select **Switch** from the list.

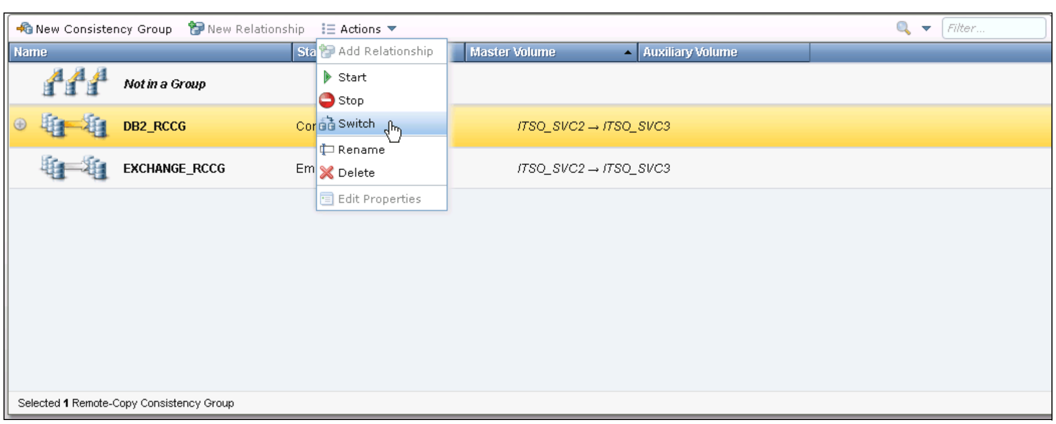


Figure 4-50 Switch the copy direction on a Consistency Group

4. A Warning window opens (Figure 4-51). A confirmation is needed to switch the Consistency Group direction. In the example shown in Figure 4-51, the Consistency Group is switched from the master group to the auxiliary group. Click **OK** to confirm your choice.

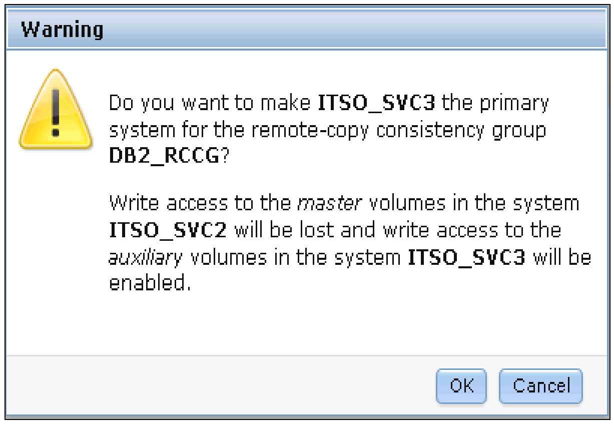


Figure 4-51 Warning window for switching the copy direction on a Consistency Group

5. The remote copy direction is now switched, as shown in Figure 4-52. The auxiliary volume is now accessible and indicated as primary volume.

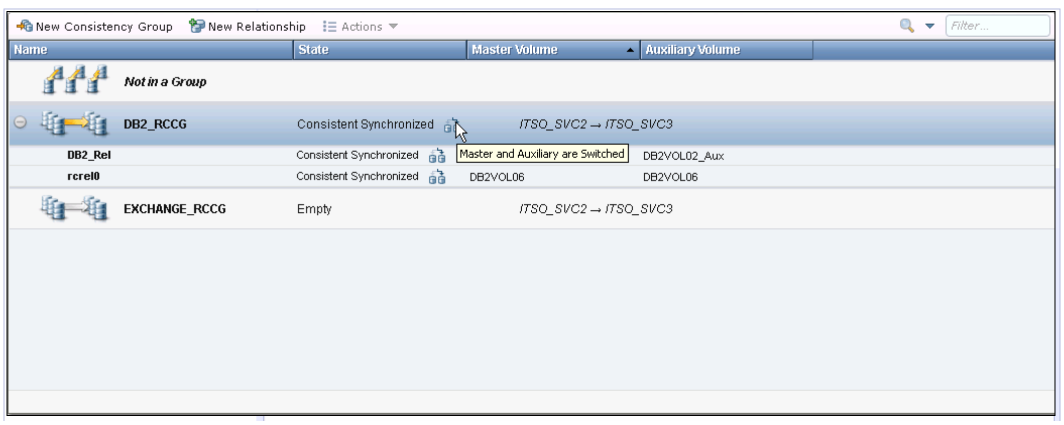


Figure 4-52 Master volume and auxiliary volume are switched

4.4.12 Stopping a Metro Mirror or Global Mirror relationship

After it is started, the Metro Mirror or Global Mirror process can be stopped. Only relationships that are not members of a Consistency Group, or relationships that are the sole members of a Consistency Group, can be stopped individually. You can also use this command to enable write access to a consistent secondary volume.

Perform the following steps to stop a remote copy relationship:

1. From the SVC Overview panel, click **Copy Services** → **Remote Copy**.
2. Expand the **Not in a Group** column.
3. In the table, select the remote copy relationship that you want to stop.
4. Click **Actions** → **Stop** (Figure 4-53) to stop the remote copy process.

> **Tip:** You can also right-click a relationship and select **Stop** from the list.

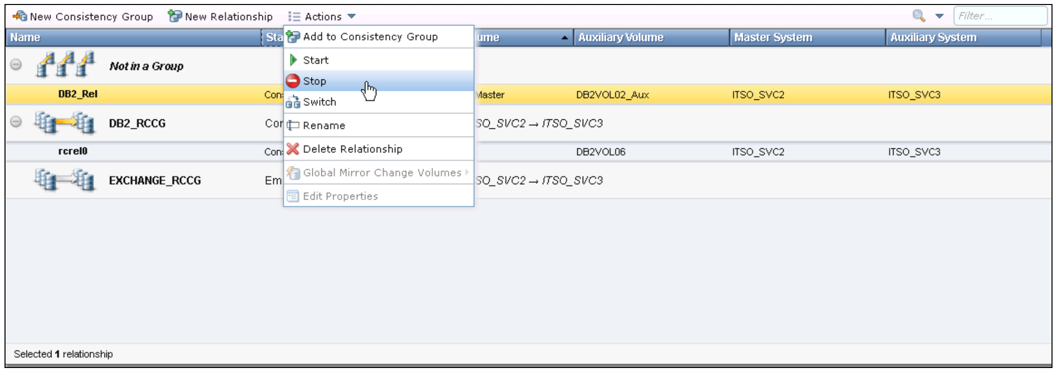


Figure 4-53 Stopping a relationship

5. The Stop Remote Copy Relationship window opens (Figure 4-54). To allow secondary read/write access, select **Allow secondary read/write access**, and then click **Stop Relationship** to confirm your choice.

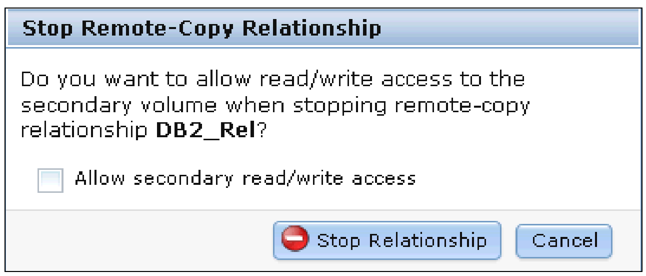


Figure 4-54 Do you want to allow access to the secondary volume?

6. The new relationship status can be checked, as shown in Figure 4-55 on page 147. The relationship is now `Consistent Stopped`.

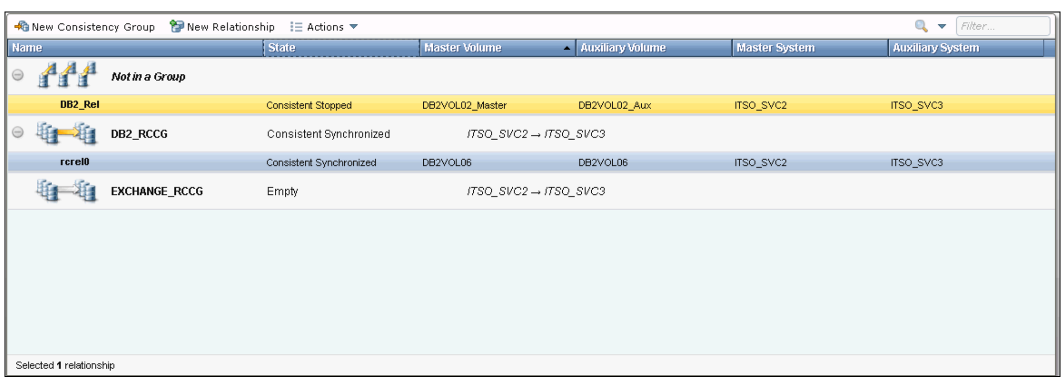


Figure 4-55 Consistent Stopped relationship - Metro or Global Mirror

4.4.13 Stopping a Consistency Group

After it is started, the Consistency Group can be stopped. You can also use this command to enable write access to consistent secondary volumes.

Perform the following steps to stop a Consistency Group:

1. From the Overview panel, click **Copy Services** → **Remote Copy**.

2. In the table, select the Consistency Group that you want to stop.
3. Click **Actions** → **Stop** (Figure 4-56) to stop the remote copy Consistency Group.

> **Tip:** You can also right-click a relationship and select `Stop` from the list.

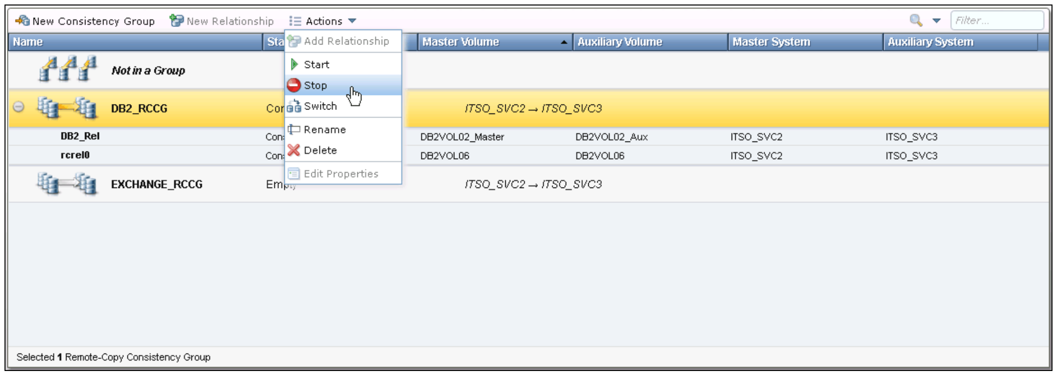


Figure 4-56 Stopping a Consistency Group

4. The Stop Remote Copy Consistency Group window opens (Figure 4-57 on page 148). To allow secondary read/write access, select **Allow secondary read/write access**, and then click **Stop Consistency Group** to confirm your choice.

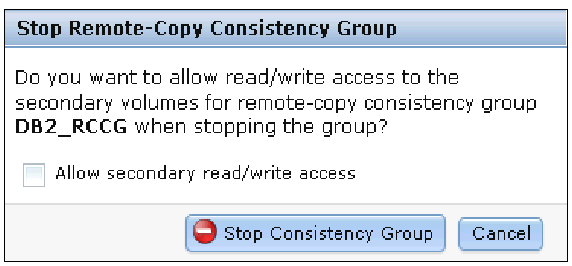


Figure 4-57 Do you want to allow access to the secondary volumes?

5. The new relationship status can be checked, as shown in Figure 4-58. The relationship is now `Consistent Stopped`.

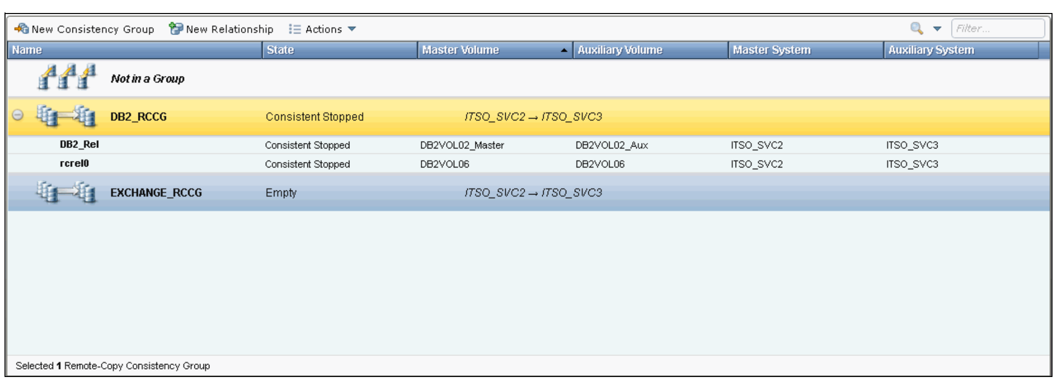


Figure 4-58 Consistent Stopped - Consistency Group

4.4.14 Deleting stand-alone relationships

Perform the following steps to delete a stand-alone remote copy mapping:

1. From the Overview panel, click **Copy Services** → **Remote Copy**.
2. In the table, select the remote copy relationship that you want to delete.

> **Tip:** To select multiple remote copy mappings, hold down the **Ctrl** key and use your mouse to select the entries that you want.

3. Click **Actions** → **Delete Relationship** (Figure 4-59 on page 149).

> **Tip:** You can also right-click a remote copy mapping and select **Delete Relationship** from the list.

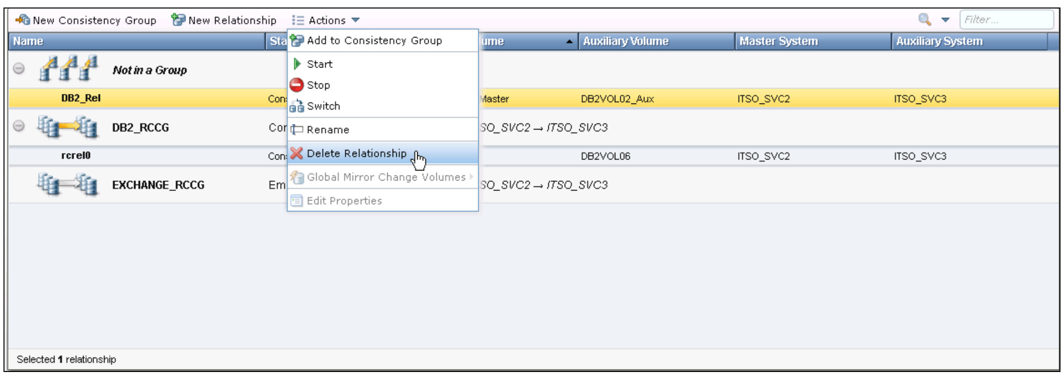


Figure 4-59 Deleting a relationship

4. The Delete Relationship window opens (Figure 4-60). In the "Verify the number of relationships that you are deleting" field, enter the number of volumes that you want to remove. This verification has been added to help you avoid deleting the wrong relationships.

   Click **Delete** to complete the operation (Figure 4-60).

Figure 4-60 Verify the relationship to delete

4.4.15 Deleting a Consistency Group

Deletion consideration: Deleting a Consistency Group does not delete the volume relationships.

Perform the following steps to delete a Consistency Group:

1. From the Overview panel, click **Copy Services** → **Remote Copy**.
2. In the left column, select the Consistency Group that you want to delete.
3. Click **Actions** → **Delete** (Figure 4-61 on page 150).

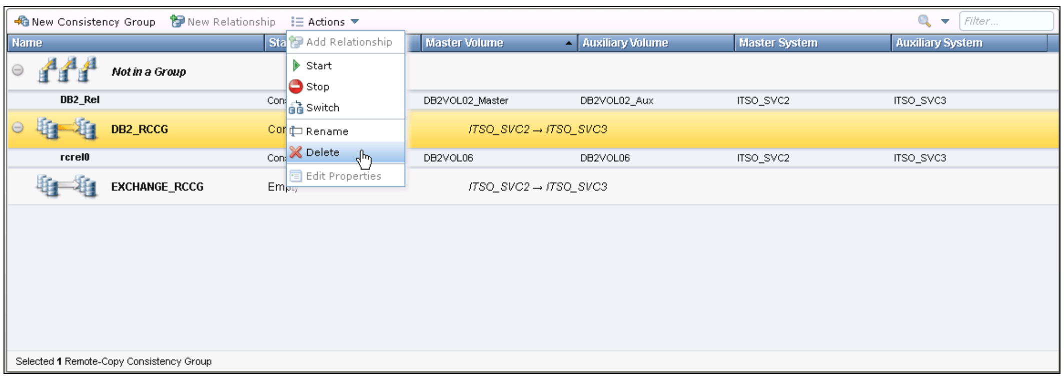


Figure 4-61 Deleting a Consistency Group

4. A Warning window opens, as shown in Figure 4-62. Click **OK** to complete the operation.

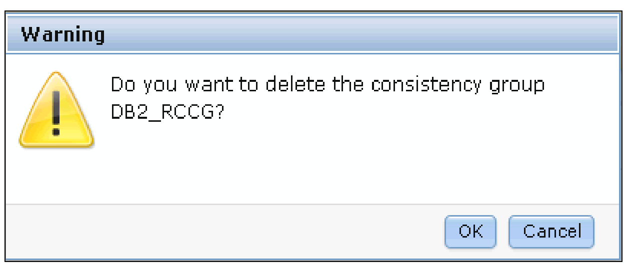


Figure 4-62 Do you want to delete the Consistency Group?

5

FlashCopy

The FlashCopy function is a point-in-time copy between source volumes and target volumes. This chapter explains the characteristics of the FlashCopy function including use and example cases, and describes how to implement it.

The following topics are discussed:

- Introduction to FlashCopy
- FlashCopy characteristics
- FlashCopy internals
- FlashCopy events and states
- Dependencies between FlashCopy mappings
- Mapping tree example

5.1 Introduction to FlashCopy

FlashCopy is an IBM product supported by many IBM Storage subsystems. With the FlashCopy function, as soon as the copy operation has started, the data on target volumes is replaced by data from source volumes. FlashCopy can be referred to by other names including Time-Zero copy, T_O, point-in-time copy, or Snapshot copy.

5.1.1 Life without the FlashCopy function

If FlashCopy is not being used and you need a consistent copy of data for a specific point in time, the application and the operating system must manipulate the I/O because the I/O needs to be “quiesced” while the copy process is running. The time required to make a consistent copy depends on factors such as the amount of data to be copied and the environment itself including Fibre Channel speed, disk speed, and usage.

The most important detail is that the data is only available on the target volume when the copy is completely finished, and this must be managed and monitored.

5.1.2 Life with the FlashCopy function

Point-in-time copy techniques are used to help solve the problem that it is difficult to make a consistent copy of a data set that is being constantly updated. If a copy of a data set is taken using technology that does not provide point-in-time semantics, and the data set changes during the copy operation, the resulting copy might contain data that is not consistent. For example, if a reference to an object is copied earlier than the object itself and the object is moved before it is itself copied, then the copy will contain the referenced object at its new location but the reference will point to the old location.

Using FlashCopy, you create a copy of source volume to a target volume, using a relationship in an extremely short time and write I/O for an application does not need to be “quiesced” for such a long time. With the FlashCopy operation the source volume is briefly “frozen” to initialize the FlashCopy bitmap, and then the I/O on the source volume is allowed to resume. The impression is that the data is already available on the target volume; however, the copy process is still running in the background. For more information about bitmaps, see 5.3.6, “bitmaps” on page 168.

As soon as the copy process is completed, the relationship can be removed and the data of the target volume is the same as the source volume from the point in time when the FlashCopy process was invoked. If you have data using multiple volumes, you can create Consistency Groups and include all volumes into the Consistency Group. This means the data on target volumes has the same data from the source volumes at the point in time that the copy started. For more information about Consistency Groups, see 5.2.3, “FlashCopy Consistency Groups” on page 155.

Using FlashCopy gives you the flexibility to have up to 256 target volumes, and makes it possible to use the Consistency Group to create different point-in-time copies from the same source volume. For more information, see 5.3.8, “FlashCopy maximum configurations” on page 169.

Also, as soon as you have a “fully” copied target volume, FlashCopy is able to update the target volume using incremental copies. This means that the data of the target volume contains only the changed data from the source volume since the FlashCopy process started. For more information, see 5.2.7, “Incremental FlashCopy” on page 158.

5.1.3 FlashCopy function use and example cases

This section discusses use and example cases for the FlashCopy function. To address your challenges, you need to adopt a combined business and technical view of the problem or problems to be solved. First determine what you need to address from a business perspective, then ascertain whether FlashCopy can meet these expectations.

Here are common use cases that utilize FlashCopy:

- Rapidly creating consistent backups of dynamically changing data
- Rapidly creating consistent copies of production data to facilitate data movement or migration between hosts
- Rapidly creating copies of production datasets for application development and testing
- Rapidly creating copies of production datasets for auditing purposes and data mining
- Rapidly creating copies of production datasets for quality assurance

Regardless of the business need, FlashCopy is extremely flexible and has a broad feature set, making it applicable to many scenarios.

One use is for backing up a consistent dataset without requiring a long backup window. Note that FlashCopy does not reduce the time to take a backup to the traditional backup infrastructure. However, it can be used to minimize and, under uncertain conditions, eliminate application downtime associated with a backup operation. After the FlashCopy operation is performed, the resulting image of the data can be backed up to tape and after finishing this operation, the data on the target volume can be discarded.

Usually when FlashCopy is used for backup purposes, the data on the target volume is accessed as “read-only” at the operating system level. This provides extra security by ensuring that the data on the target volume has not been modified and is the same as the source volume.

Another use of FlashCopy as part of a business continuity strategy is to create a full consistent copy of production data for a specific point in time at a remote location. You can combine Global Mirror/Metro Mirror and FlashCopy and in this case, take a FlashCopy from the GM/MM secondary volume.

At the second location, you can take a consistent backup of production data or create a clone of the data so that is available if anything happens to the production data.

You can also use FlashCopy to create clones of data for application development testing or for application integration testing. FlashCopy is further useful when a set of data has to be used for different purposes, for example, using a FlashCopy database for data mining.

5.2 FlashCopy characteristics

To use it successfully, it is important to understand the characteristics of FlashCopy. Several functions increase the flexibility and usefulness of FlashCopy, including those listed here.

- FlashCopy mapping
- FlashCopy Consistency Groups
- Multiple Target FlashCopy (MTFC)
- Cascaded FlashCopy (CFC)
- Incremental FlashCopy (IFC)
- Thin-provisioned FlashCopy

These characteristics are explained in more detail in the following sections.

5.2.1 FlashCopy mapping

Keep these considerations in mind regarding FlashCopy mapping:

- The FlashCopy operation must occur between a source volume and a target volume of the same size.
- It is not possible to use FlashCopy to copy only part of a volume.
- The source and target volumes must belong to the same SAN Volume Controller Cluster. Otherwise, you need to use Metro Mirror or Global Mirror.
- The source and target volumes can be in a different I/O Group or Storage Pool.
- Volumes that are members of a FlashCopy mapping cannot have their size increased or decreased while they are members of the FlashCopy mapping.

Figure 5-1 illustrates the concept of FlashCopy mapping.

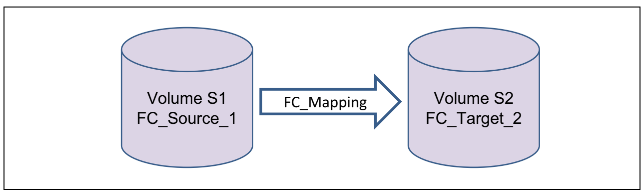


Figure 5-1 Single volume mapping

The act of creating a relationship between a source volume and a target volume is called a FlashCopy mapping. FlashCopy mappings can be either stand-alone or a member of a Consistency Group. You can perform the actions of preparing, starting, or stopping FlashCopy on either a stand-alone mapping or a Consistency Group. You need to associate a name to the mapping. This is considered as an alias to manage the source and target volumes pairs during the FlashCopy operation instead of using the names of the volumes themselves.

In the FlashCopy mapping, the source volume provides the data to be copied and the target volume represents the data of the source volume for the point in time when the FlashCopy operation was started. After this copy is completed, the FlashCopy mapping is no longer needed for the target volume to access the data.

You can set the autodelete option to “on” or “off” while creating or modifying a new FlashCopy mapping. Setting autodelete to “on” means that the FlashCopy mapping is deleted automatically as soon as the background copy has finished copying all data to the target volume. This is especially useful in scripts. Also, deleting mappings frees bitmap space.

5.2.2 Number of FlashCopy mappings in a cluster

The maximum number of FlashCopy mappings is 4096 for the entire cluster. Each single mapping (which might be part of an MTFC or a CFC) contributes to the maximum number of mappings.

5.2.3 FlashCopy Consistency Groups

Consistency Groups can be used to help create a consistent point-in-time copy across multiple volumes, thus managing the consistency of dependent writes that are executed in the application following the correct sequence.

When Consistency Groups are used, the FlashCopy commands are issued to the Consistency Groups, which perform the operation on all FlashCopy mappings contained within the Consistency Groups at the same time.

Figure 5-2 illustrates a Consistency Group consisting of two volume mappings.

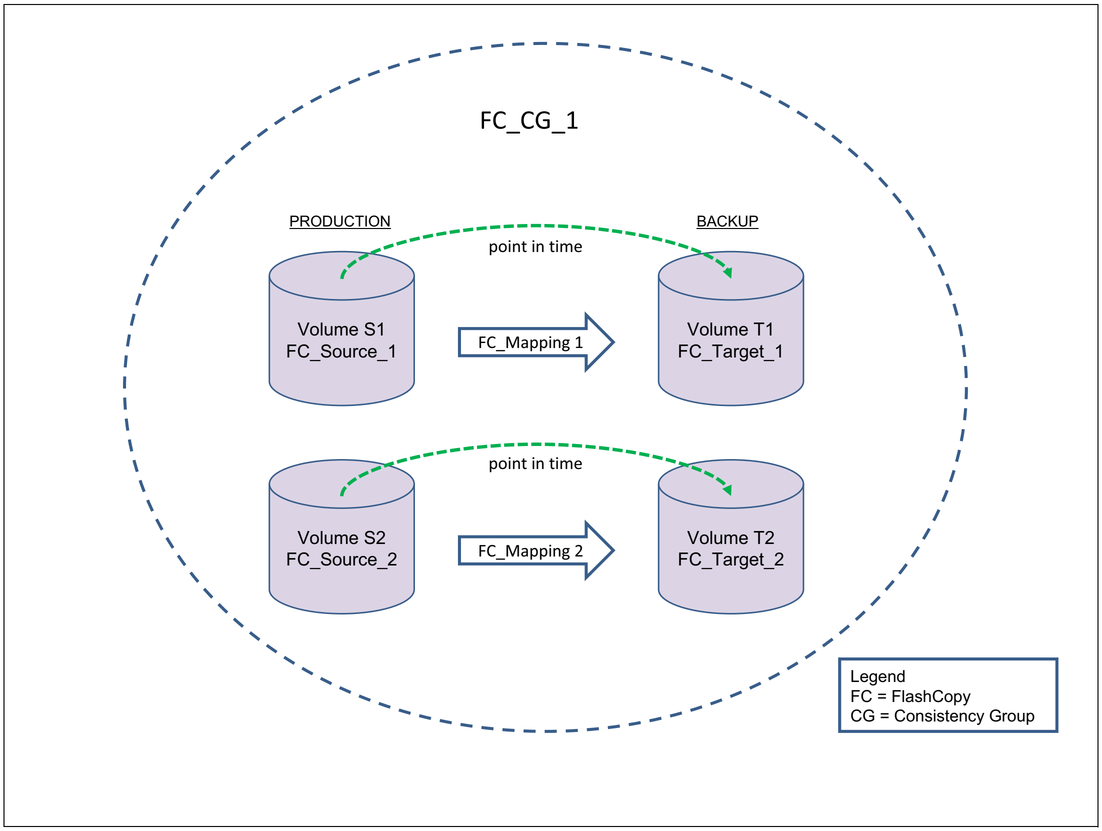


Figure 5-2 Multiple volume mapping in a Consistency Group

> **FlashCopy mapping considerations:** If the FlashCopy mapping has been added to a Consistency Group, it can only be managed as part of the group. This also means that FlashCopy operations such as **`prepare`**, **`start`** and **`stop`** are no longer allowed on the individual FlashCopy mappings.

Dependent writes

As an example for Consistency Group use, some data that must be consistent can span multiple volumes, such as when database management system (DBMS) logs reside on different volumes than the database. This introduces the concept of *dependent writes*, which need to be accounted for. Dependent writes occur when database logs are written before (to indicate a database update is about to take place) and after (to log a successful write operation to the database) the database gets updated.

If you start a FlashCopy operation of the associated volumes without ensuring consistency across the volumes, the FlashCopy target can contain both written logs but *not* the database update. The logs in the FlashCopy target indicate that the write I/O has occurred, but the actual write I/O is missing from the target volumes that holds the database. This renders the FlashCopy target data inconsistent, from a DBMS point of view.

A Consistency Group may be created with, or be modified to have, the **`autodelete`** option specified. This makes the Consistency Group delete itself when the last mapping in the group is deleted or moved out of the Consistency Group.

For example, if you have a Consistency Group with two FlashCopy mappings and you remove both mappings, the Consistency Group is automatically deleted from the cluster configuration. This can improve administration time if the Consistency Group is used only one time.

5.2.4 Multiple Target FlashCopy

In Multiple Target FlashCopy (MTFC), a source volume can be used in multiple FlashCopy mappings, while the target is a different volume, as shown in Figure 5-3.

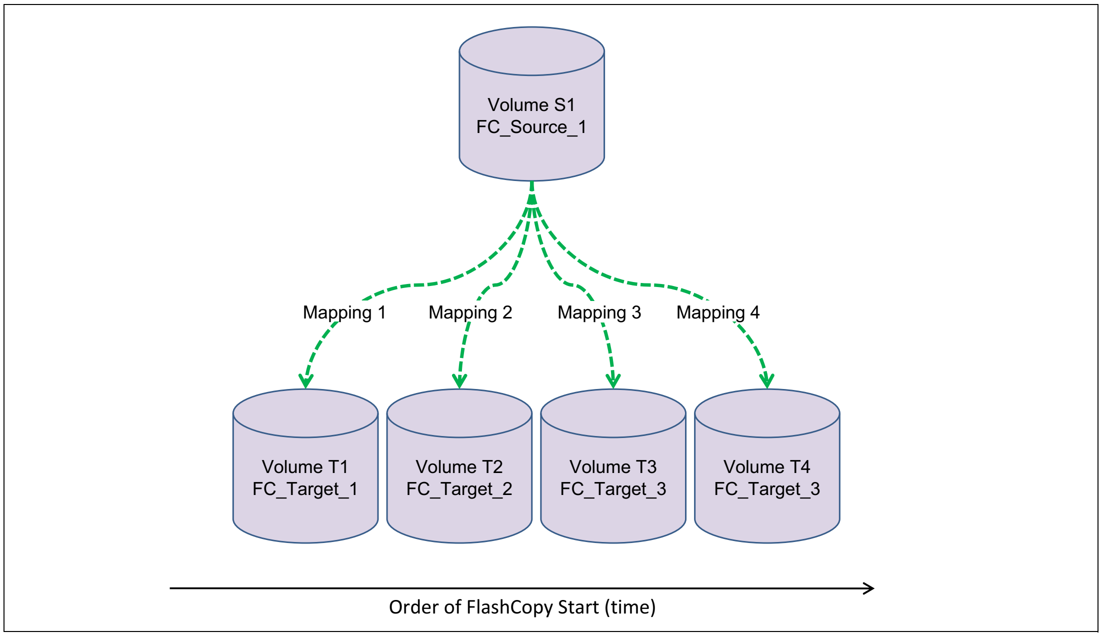


Figure 5-3 Multiple Target FlashCopy (MTFC)

Up to 256 different mappings are possible for each source volume. These mappings are independently controllable from each other. For instance, the background copy priority can be different for these mappings, as explained in 5.2.9, “Background copy” on page 161. The implementation of MTFC introduces a few dependencies, as discussed in 5.5, “Dependencies between FlashCopy mappings” on page 175.

MTFC mappings can be members of the same or different Consistency Groups. In cases where all the mappings are in the same Consistency Group, the result of starting the Consistency Group will be to multiple identical target volumes.

5.2.5 Cascaded FlashCopy

With Cascaded FlashCopy (CFC), you can have a source volume for one FlashCopy mapping and as the target for another FlashCopy mapping; this is referred to as a *Cascaded FlashCopy*. This function is illustrated in Figure 5-4.

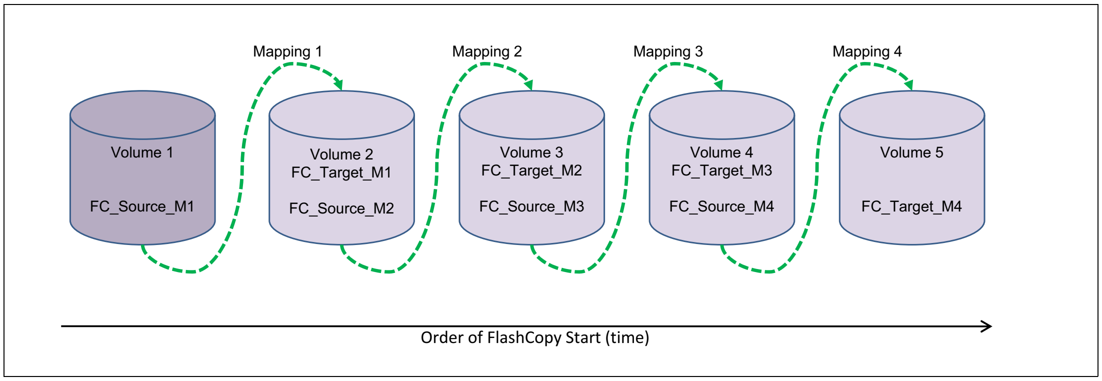


Figure 5-4 Cascaded FlashCopy

A total of 256 mappings are possible for each cascade; Figure 5-4 shows only four FlashCopy mappings. See more details about the dependencies between FlashCopy mappings in 5.5, "Dependencies between FlashCopy mappings" on page 175.

Be aware that the use of Consistency Groups is restricted when using CFC. A Consistency Group serves the purpose of invoking FlashCopy mappings at the same point in time. Within the *same* Consistency Group, it is not possible to have mappings where:

- The source volume of one mapping is the target of another mapping.
- The target volume of one mapping is the source volume for another mapping.

These combinations are not useful because within a Consistency Group, mappings cannot be established in a certain order, which renders the content of the target volume undefined. For instance, it is not possible to determine whether the first mapping was established before the target volume of the first mapping that acts as a source volume for the second mapping.

Even if it were possible to ensure the order in which the mappings are established within a Consistency Group, the result is equal to MTFC (that is, two volumes holding the same target data for one source volume). In other words, a cascade is useful for copying volumes in a certain order (and copying the changed content targets of FlashCopies), rather than at the same time in an undefined order (from within one single Consistency Group).

5.2.6 Combined Multiple Target FlashCopy and Cascaded FlashCopy

MTFC and CFC can be combined in any way if the combination of both copy methods does not exceed 16 mappings in one tree. Combining MTFC and CFC is illustrated in Figure 5-5 on page 158.

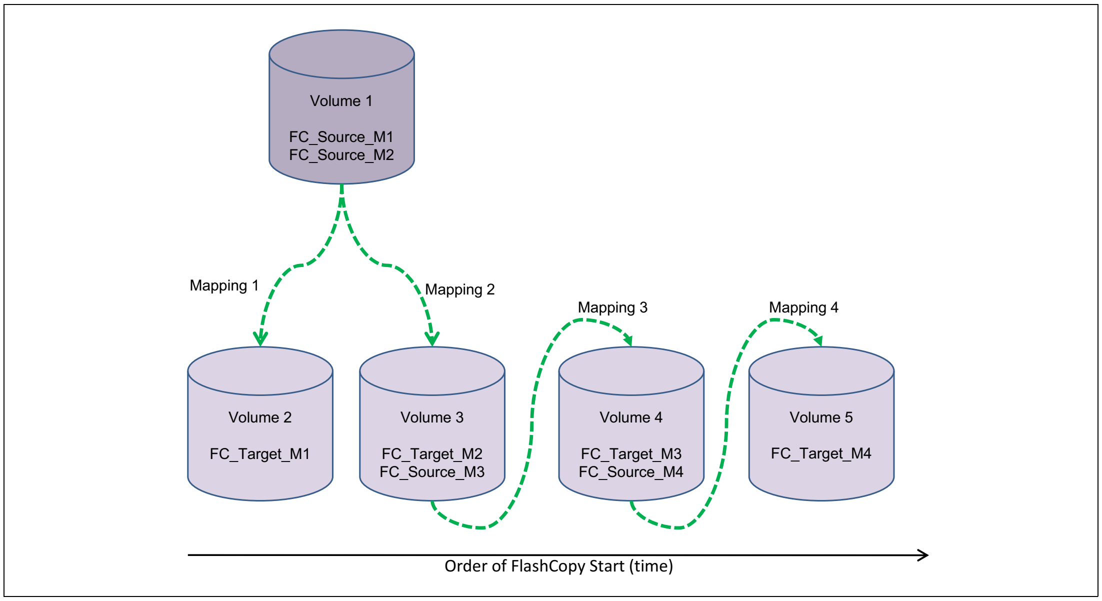


Figure 5-5 A tree of combined MTFC and CFC

As shown in Figure 5-5, a “target volume” can be the source, the target, or a source *and* a target if it is member of one or more FlashCopy mappings.

5.2.7 Incremental FlashCopy

Using Incremental FlashCopy (IFC), you can reduce the required time of copy. Also, because less data has to be copied, the workload put on the cluster and the back-end disk subsystems has been reduced.

Basically, Incremental FlashCopy (IFC) does not require that you copy an entire disk source volume every time the FlashCopy mapping is started. It means that only the changed regions on source volumes are copied to target volumes, as shown in Figure 5-6 on page 159.

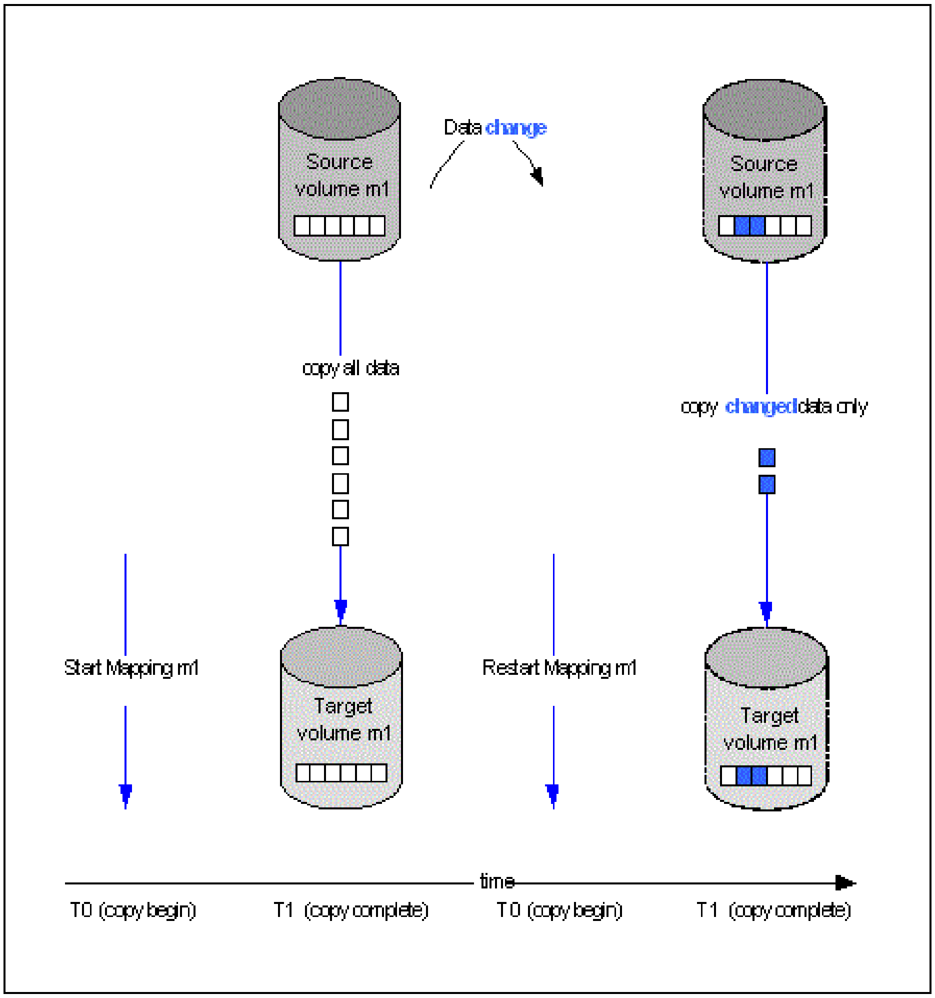


Figure 5-6 Incremental FlashCopy (IFC)

If the FlashCopy mapping was stopped before the background copy had completed, then when the mapping is restarted, the data that had been copied before the mapping was stopped will not be copied again. For example, if an incremental mapping reaches 10 percent progress when it is stopped and then it is restarted, that 10 percent of data will not be recopied when the mapping is restarted, assuming that it was not changed.

Stopping an incremental FlashCopy mapping: If you are planning to stop an incremental FlashCopy mapping, make sure that the copied data on the source volume will not be changed, if possible. Otherwise, you can have an inconsistent point-in-time copy.

The amount of data to be copied when refreshing an incremental mapping depends on the grain size, as well. If the grain size is 64 KB, as compared to 256 KB, there might be less data to copy to get a fully independent copy of the source again.

A “difference” value is provided in the query of a mapping, which makes it possible to know how much data has changed. This data must be copied when the Incremental FlashCopy mapping is restarted. The difference value is the percentage (0-100 percent) of grains that have been changed, and it must be copied to the target volume to get a fully independent copy of the source volume.

The functionality of Incremental FlashCopy is useful in cases where you want a full copy of a volume, for example, to be used by application testing or data mining, and also want to refresh this copy on a regular basis.

When clients want to maintain fully independent copies of data as part of their disaster tolerance strategy, Incremental FlashCopy makes it possible to refresh these copies often, without using too much time and resource.

However, by using incremental FlashCopy, you use up to double the bitmap space as compared to FlashCopy without using the incremental.

5.2.8 Thin-provisioned FlashCopy

When a new volume is created, you can designate it as a *thin-provisioned volume*, and it has a virtual capacity and a real capacity.

*Virtual capacit*y is the volume storage capacity that is available to a host. *Real capacity* is the storage capacity that is allocated to a volume copy from a storage pool. In a fully allocated volume, the virtual capacity and real capacity are the same. In a thin-provisioned volume, however, the virtual capacity can be much larger than the real capacity.

The virtual capacity of a thin-provisioned volume is typically larger than its real capacity. On SAN Volume Controller and Storwize V7000 systems, the real capacity is used to store data that is written to the volume, and metadata that describes the thin-provisioned configuration of the volume. As more information is written to the volume, more of the real capacity is used.

Thin-provisioned volumes can also help to simplify server administration. Instead of assigning a volume with some capacity to an application and increasing that capacity following the needs of the application if those needs change, you can configure a volume with a large virtual capacity for the application and then increase or shrink the real capacity as the application needs change, without disrupting the application or server.

When you configure a thin-provisioned volume, you can use the warning level attribute to generate a warning event when the used real capacity exceeds a specified amount or percentage of the total real capacity. For example, if you have a volume with 10 GB of total capacity and you set the warning to 80 percent, an event will be registered in the event log when you use 80 percent of the total capacity. This is useful when you need to control how much of the volume is used.

If a thin-provisioned volume does not have enough real capacity for a write operation, the volume is taken offline and an error is logged (error code 1865, event ID 060001). Access to the thin-provisioned volume is restored by either increasing the real capacity of the volume or increasing the size of the storage pool on which it is allocated.

You can use thin volumes for cascaded FlashCopy and multiple target FlashCopy. It is also possible to mix thin-provisioned with normal volumes. It can be used for incremental FlashCopy too, but using thin-provisioned volumes for incremental FlashCopy only makes sense if the source and target are thin-provisioned.

Performance consideration: The best performance is obtained when the grain size of the thin-provisioned volume is the same of the grain size of the FlashCopy mapping.

The recommendation for thin-provisioned FlashCopy:

- Thin-provisioned volume grain size must be equal to the FlashCopy grain size.
- Thin-provisioned volume grain size must be 64 KB for the best performance and the best space efficiency.

The exception is where the thin target volume is going to become a production volume (and is likely to be subjected to ongoing heavy I/O). In this case, the 256 KB thin-provisioned grain size is recommended to provide better long term I/O performance at the expense of a slower initial copy.

> **FlashCopy grain size considerations:** Even if the 256 KB thin-provisioned volume grain size is chosen, it is still beneficial to limit the FlashCopy grain size to 64 KB. It is possible to minimize the performance impact to the source volume, even though this size increases the I/O workload on the target volume.
>
> However, clients with extremely large numbers of FlashCopy/Remote Copy relationships might still be forced to choose a 256 KB grain size for FlashCopy to avoid constraints on the amount of bitmap memory.

Thin-provisioned incremental FlashCopy

The implementation of thin-provisioned volumes does not preclude the use of incremental FlashCopy on the same volumes. It does not make sense to have a fully allocated source volume and then use incremental FlashCopy, which is always a full copy at first time, to copy this fully allocated source volume to a thin-provisioned target volume. However, this action is not prohibited.

Consider this optional configuration:

- A thin-provisioned source volume can be copied incrementally using FlashCopy to a thin-provisioned target volume. Whenever the FlashCopy is performed, only data that has been modified is recopied to the target. Note that if space is allocated on the target because of I/O to the target volume, this space will not be reclaimed with subsequent FlashCopy operations.
- A fully allocated source volume can be copied incrementally using FlashCopy to another fully allocated volume at the same time as it is being copied to multiple thin-provisioned targets (taken at separate points in time). This combination allows a single full backup to be kept for recovery purposes, and separates the backup workload from the production workload. At the same time, it allows older thin-provisioned backups to be retained.

5.2.9 Background copy

With *FlashCopy background copy* enabled, the source volume data will be copied to the corresponding target volume. A value of 0 disables the background copy. If the FlashCopy background copy is disabled, only data that has changed on the source volume will be copied to the target volume.

The benefit of using a FlashCopy mapping with background copy enabled is that the target volume becomes a real clone (independent from the source volume) of the FlashCopy mapping source volume after the copy is complete. When the background copy function is not performed, the target volume only remains a valid copy of the source data while the FlashCopy mapping remains in place.

The *background copy rate* is a property of a FlashCopy mapping that is defined as a value between 0 and 100. The background copy rate can be defined and changed dynamically for individual FlashCopy mappings.

Table 5-1 shows the relationship of the background copy rate value to the attempted number of grains to be copied per second.

Table 5-1 Relationship between the rate, data rate and grains per second

Value	Data copied per second	Grains per second (256 KB grain)	Grains per second (64 KB grain)
1 - 10	128 KB	0.5	2
11 - 20	256 KB	1	4
21 - 30	512 KB	2	8
31 - 40	1 MB	4	16
41 - 50	2 MB	8	32
51 - 60	4 MB	16	64
61 - 70	8 MB	32	128
71 - 80	16 MB	64	256
81 - 90	32 MB	128	512
91 - 100	64 MB	256	1024

The grains per second numbers represent the maximum number of grains that will be copied per second, assuming that the bandwidth to the storage pool can accommodate this rate.

If it is unable to achieve these copy rates because of insufficient bandwidth from nodes to the storage pool, the background copy I/O contends for resources on an equal basis with the I/O that is arriving from the hosts. Both background copy and foreground I/O continue to make forward progress, and do not stop, hang, or cause the node to fail. The background copy is performed by both nodes of the I/O Group in which the source volume is located.

Figure 5-7 on page 163 illustrates the background copy throughput capabilities.

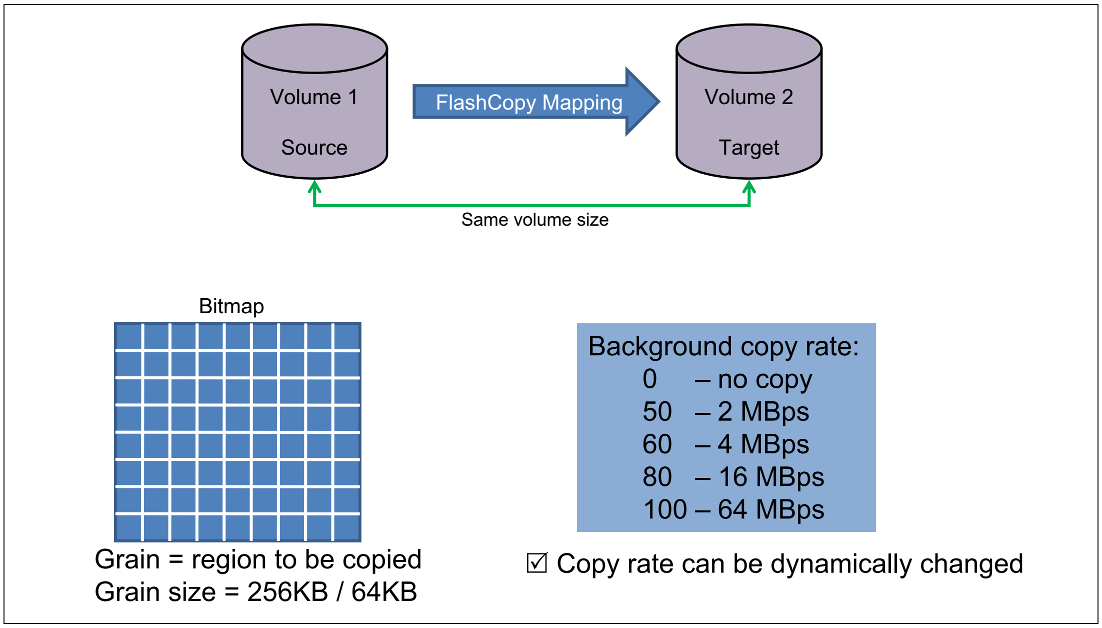


Figure 5-7 Background copy performance details

5.3 FlashCopy internals

Understanding how FlashCopy works internally helps you to configure it in a way that you want and enables you to obtain more benefits from it.

5.3.1 Indirection layer

The *FlashCopy indirection layer* governs the I/O to the source and target volumes when a FlashCopy mapping is started, which is done using a FlashCopy bitmap. The purpose of the FlashCopy indirection layer is to enable both the source and target volumes for read and write I/O immediately after the FlashCopy has been started.

The following description illustrates how the FlashCopy indirection layer works when a FlashCopy mapping is prepared and then started.

When a FlashCopy mapping is prepared and started, the following sequence is applied:

1. Flush the write cache to the source volume or volumes that are part of a Consistency Group.
2. Put the cache into write-through mode on the source volumes.
3. Discard the cache for the target volumes.
4. Establish a sync point on all of the source volumes in the Consistency Group (creating the FlashCopy bitmap).
5. Ensure that the indirection layer governs all of the I/O to the source volumes and target.
6. Enable the cache on source volumes and target volumes.

FlashCopy provides the semantics of a point-in-time copy using the indirection layer, which intercepts I/O that is directed at either the source or target volumes. The act of starting a

FlashCopy mapping causes this indirection layer to become active in the I/O path, which occurs automatically across all FlashCopy mappings in the Consistency Group. The indirection layer then determines how each of the I/O is to be routed based on the following factors:

- The volume and the logical block address (LBA) to which the I/O is addressed
- Its direction (read or write)
- The state of an internal data structure, the FlashCopy bitmap

The indirection layer allows the I/O to go through the underlying volume; redirects the I/O from the target volume to the source volume; or queues the I/O while it arranges for data to be copied from the source volume to the target volume. To explain which action is applied for each I/O, we first look at the FlashCopy bitmap; refer to 5.3.6, “bitmaps” on page 168 for more information about this topic.

5.3.2 Summary of FlashCopy indirection layer algorithm

Table 5-2 summarizes the indirection layer algorithm.

Table 5-2 Summary table of the FlashCopy indirection layer algorithm

Volume being accessed	**Has the grain been copied?**	**Host I/O operation**	
		Read	**Write**
Source	No	Read from the source volume.	Copy grain to the most recently started target for this source, then write to the source.
	Yes	Read from the source volume.	Write to the source volume.
Target	No	If any newer targets exists for this source in which this grain has already been copied, read from the oldest of these targets. Otherwise, read from the source.	Hold the write. Check the dependency target volumes to see if the grain has been copied. If the grain is not already copied to the next oldest target for this source, copy the grain to the next oldest target. Then, write to the target.
	Yes	Read from the target volume.	Write to the target volume.

5.3.3 Discussion of grains

When data is copied between volumes, it is copied in units of address space known as *grains*. Grains are units of data grouped together to optimize the use of the bitmap that keeps track of changes to the data between the source and target volume. You have the option of using 64 KB or 256 KB grain sizes; 256 KB is the default (starting with SVC code level 6.4).

The FlashCopy bitmap contains one bit for each grain and is used to keep track of whether the source grain has been copied to the target. Note that the 64 KB grain size consumes bitmap space at a rate of four times the default 256 KB size.

The FlashCopy bitmap dictates read and write behavior for both the source and target volumes, as explained here.

Source reads

Reads are performed from the source volume, which is the same as for non-FlashCopy volumes.

Source writes

Writes to the source will cause the grain to be copied to the target if it has not already been copied, the bitmap to be updated, and then the write will be performed to the source.

Target reads

Reads are performed from the target if the grain has already been copied. Otherwise, the read is performed from the source and no copy is performed.

Target writes

Writes to the target will cause the grain to be copied from the source to the target unless the entire grain is being updated on the target. In this case, the target will be marked as split with the source (if there is no I/O error during the write) and the write will go directly to the target.

5.3.4 FlashCopy indirection layer algorithm

Imagine the FlashCopy indirection layer as the I/O traffic director when a FlashCopy mapping is active. The I/O is intercepted and handled according to whether it is directed at the source volume or at the target volume, depending on the nature of the I/O (read or write) and the state of the grain (whether it has been copied).

Figure 5-8 on page 166 illustrates how the background copy runs while I/Os are handled according to the indirection layer algorithm.

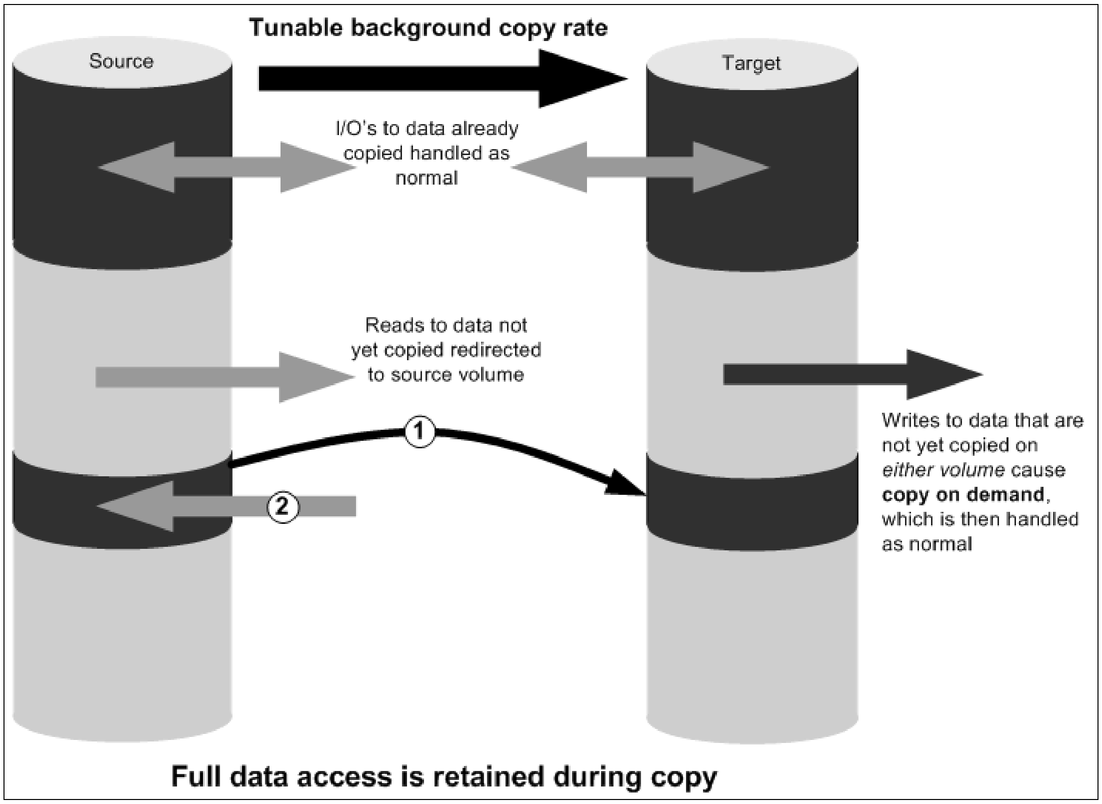


Figure 5-8 I/O processing with FlashCopy

5.3.5 Interaction and dependency between Multiple Target FlashCopy mappings

Figure 5-9 on page 167 represents a set of four FlashCopy mappings that share a common source. The FlashCopy mappings will target volumes Target 0, Target 1, Target 2, and Target 3.

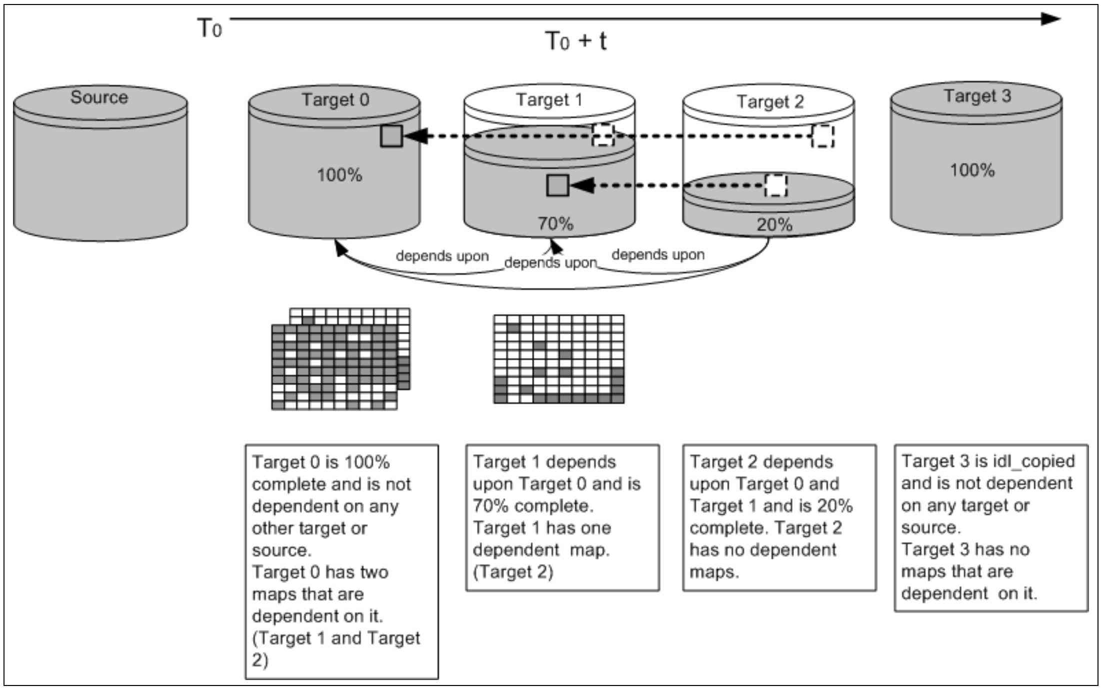


Figure 5-9 Interactions between MTFC mappings

- Target 0 is not dependent on a source, because it has completed copying. Target 0 has two dependent mappings (Target 1 and Target 2).
- Target 1 is dependent upon Target 0. It will remain dependent until all of Target 1 has been copied. Target 2 is dependent on it, because Target 2 is 20% copy complete. After all of Target 1 has been copied, it can then move to the idle_copied state.
- Target 2 is dependent upon Target 0 and Target 1 and will remain dependent until all of Target 2 has been copied. No target is dependent on Target 2, so when all of the data has been copied to Target 2, it can move to the idle_copied state.
- Target 3 has actually completed copying, so it is not dependent on any other maps.

Target writes with Multiple Target FlashCopy

A write to an intermediate or newest target volume must consider the state of the grain within its own mapping, and the state of the grain of the next oldest mapping:

- If the grain of the next oldest mapping has not been copied yet, it must be copied before the write is allowed to proceed to preserve the contents of the next oldest mapping. The data that is written to the next oldest mapping comes from a target or source.
- If the grain in the target being written has not yet been copied, the grain is copied from the oldest already copied grain in the mappings that are newer than the target, or the source if none are already copied. After this copy has been done, the write can be applied to the target.

Target reads with Multiple Target FlashCopy

If the grain being read has already been copied from the source to the target, the read simply returns data from the target being read. If the grain has not been copied, each of the newer mappings is examined in turn and the read is performed from the first copy found. If none are found, the read is performed from the source.

5.3.6 bitmaps

A *bitmap* is an internal data structure stored in a particular I/O Group that is used to track which grains in FlashCopy mappings have been copied from the source volume to the target volume (that is, which grains have been "split"). One bit in each bitmap represents the state of one grain, and either the bitmap is split or it is not.

For Volume Mirroring and the Replication Family (FlashCopy, Metro Mirror, and Global Mirror), memory is traded against memory that is available to the cache. The amount of memory can be decreased or increased. Consider the following memory sizes when you are changing the I/O Group configuration:

- The default memory size for FlashCopy is 20 MB.
- The default memory size for Metro Mirror and Global Mirror is 20 MB.
- The default memory size for mirrored volumes is 20 MB.
- The maximum memory size that can be specified for FlashCopy is 512 MB.
- The maximum memory size that can be specified for Metro Mirror and Global Mirror is 512 MB.
- The maximum memory size that can be specified for mirrored volumes is 512 MB.
- The maximum combined memory size across all features is 552 MB.

A FlashCopy bitmap takes up the bitmap space in the memory of the cluster. The maximum size of the bitmap space is 512 MB per I/O Group, which has to be shared between FlashCopy bitmaps, Remote Copy bitmaps, and Volume Mirroring synchronization bitmaps. You assign the bitmap space to the copy functions by using a variable.

This enables you to trade off memory among the cache, FlashCopy, Remote Copy (Metro Mirror and Global Mirror), and Volume Mirroring.

5.3.7 Usable bitmap space

Table 5-3 illustrates the relationship of bitmap space to FlashCopy address space, depending on the size of the grain and the kind of copy service is being used.

Table 5-3 Relationship of bitmap space to FlashCopy address space for the specified I/O Group

Copy Service	Grain size in KB	1 MB of memory provides the following volume capacity for the specified I/O Group
Metro Mirror and Global Mirror	256	2 TB of volume capacity
FlashCopy	256	2 TB of target volume capacity
FlashCopy	64	512 GB of target volume capacity
Incremental FlashCopy	256	1 TB of target volume capacity
Incremental FlashCopy	64	256 GB of target volume capacity

Copy Service	Grain size in KB	1 MB of memory provides the following volume capacity for the specified I/O Group
Volume Mirroring	256	2 TB of mirrored volume capacity

Mapping consideration: For multiple FlashCopy targets, you must consider the number of mappings. For example, for a mapping with a 256 KB grain size, 8 KB of memory allows one mapping between a 16 GB source volume and a 16 GB target volume. Alternatively, for a mapping with a 256 KB grain size, 8 KB of memory allows two mappings between one 8 GB source volume and two 8 GB target volumes.

When you create a FlashCopy mapping, if you specify an I/O Group other than the I/O Group of the source volume, the memory accounting goes towards the specified I/O Group, not towards the I/O Group of the source volume.

5.3.8 FlashCopy maximum configurations

To plan for and implement FlashCopy, you must check the configuration limits and adhere to them. Table 5-4 shows the limits for a cluster.

Table 5-4 FlashCopy properties and maximum configurations

FlashCopy property	Maximum	Comment
FlashCopy targets per source	256	This maximum is the maximum number of FlashCopy mappings that can exist with the same source volume.
FlashCopy mappings per cluster	4096	The number of mappings is no longer limited by the number of volumes in the cluster, so the FlashCopy component limit applies.
FlashCopy Consistency Groups per cluster	127	This maximum is an arbitrary limit that is policed by the software.
FlashCopy volume space per I/O Group	1024 TB	This maximum is a limit on the quantity of FlashCopy mappings using bitmap space from this I/O Group. This maximum configuration will consume all 512 MB of bitmap space for the I/O Group and allow no Metro and Global Mirror bitmap space. The default is 40 TB.
FlashCopy mappings per Consistency Group	512	This limit is due to the time that is taken to prepare a Consistency Group with a large number of mappings.

5.3.9 Summary of FlashCopy function rules and characteristics

To summarize FlashCopy rules and characteristics:

- FlashCopy is a copy process between two volumes associated with a FlashCopy mapping (see more details in 5.2.1, “FlashCopy mapping” on page 154).
- When used in a FlashCopy mapping, a volume becomes a source or target volume.
- A FlashCopy copies an entire source volume data to a target volume within one cluster.

- The volume chosen as the source and target volume can be provided by a different I/O Group or Groups.
- A volume can be the source volume in multiple FlashCopy mappings.
- A volume can be the target volume in one FlashCopy mapping.
- A volume can be the source in one mapping and the target in another mapping at the same time.
- The content of the source volume does not get changed by the FlashCopy process.
- The original content of the target volume is destroyed by the FlashCopy process.
- After the FlashCopy is established, the target volume represents a clone of the source volume.
- After the FlashCopy is established, the target volume can be accessed as read/write.
- Until all of the data is copied, the target volume presents the data as long as the mapping exists.
- After all of the data is copied, you can delete the mapping, and the volume is independent again.
- After all of the data is copied, the former target volume holds the data but without the mapping.
- To maintain data integrity, ensure that all application I/Os and host I/Os are flushed from any application and operating system buffers.
- You might need to stop your application for it to be “restarted” with a copy of the volume that has been made. Check with your application vendor if you have any doubts.
- Be careful if you want to map the target flash-copied volume to the same host that already has the source volume mapped to it. Check whether your operating system supports this configuration.
- The target volume must be the same size as the source volume. However, the target volume can be a different type (image, striped, or sequential mode) or have different cache settings (cache-enabled or cache-disabled).
- The size of volumes that are members in a FlashCopy mapping cannot be changed.
- If you stop a FlashCopy mapping or a Consistency Group before it has completed, you will lose access to the target volumes. If the target volumes are mapped to hosts, they will have I/O errors.
- A volume cannot be a source in one FlashCopy mapping and a target in another FlashCopy mapping.
- A volume can be the source for up to 256 targets.
- Starting with SVC code level 6.2.0.0, a FlashCopy mapping can be created using a target volume that is part of a Remote Copy relationship. This enables the reverse feature to be used in conjunction with a disaster recovery implementation. It also enables fast failback from a consistent copy held on a FlashCopy target volume at the auxiliary cluster to the master copy.
- FlashCopy services can only be provided inside a cluster. If you want to FlashCopy to remote storage, the remote storage needs to be defined locally to the cluster.
- Be careful if you want to map the target volume to the same host that already has the source volume mapped to it. Check that your operating system supports this configuration.

5.4 FlashCopy events and states

FlashCopy events that modify the states of FlashCopy mapping are explained here. The mapping events are listed and described in Table 5-5 on page 171.

The following list provides an overview of a FlashCopy sequence of events:

1. Associate the source data set with a target location (one or more source and target volumes).
2. Create a FlashCopy mapping for each source volume to the corresponding target volume. The target volume must be equal in size to the source volume.
3. Discontinue access to the target (application dependent).
4. Prepare (pre-trigger) the FlashCopy:
   a. Flush the cache for the source.
   b. Discard the cache for the target.
5. Start (trigger) the FlashCopy:
   a. Pause I/O (briefly) on the source.
   b. Resume I/O on the source.
   c. Start I/O on the target.

Table 5-5 Mapping events

Mapping event	Description
Create	A new FlashCopy mapping is created between the specified source volume and the specified target volume. The operation fails if any one of the following conditions is true: ► For code level software version 4.1.0 or earlier, the source or target volume is already a member of a FlashCopy mapping. ► For code level software version 4.2.0 or later, the source or target volume is already a target volume of a FlashCopy mapping. ► For code level software version 4.2.0 or later, the source volume is already a member of 16 FlashCopy mappings. ► For code level software version 4.3.0 or later, the source volume is already a member of 256 FlashCopy mappings. ► The node has insufficient bitmap memory. ► The source and target volume sizes differ.
Prepare	The **prestartfcmap** or **prestartfcconsistgrp** command is directed to either a Consistency Group for FlashCopy mappings that are members of a normal Consistency Group or to the mapping name for FlashCopy mappings that are stand-alone mappings. The **prestartfcmap** or **prestartfcconsistgrp** command places the FlashCopy mapping into the Preparing state. **Important**: The **prestartfcmap** or **prestartfcconsistgrp** command can corrupt any data that previously resided on the target volume, because cached writes are discarded. Even if the FlashCopy mapping is never started, the data from the target might have changed logically during the act of preparing to start the FlashCopy mapping.
Flush done	The FlashCopy mapping automatically moves from the Preparing state to the Prepared state after all cached data for the source is flushed and all cached data for the target is no longer valid.

Mapping event	Description
Start	When all of the FlashCopy mappings in a Consistency Group are in the Prepared state, the FlashCopy mappings can be started. To preserve the cross-volume Consistency Group, the start of all of the FlashCopy mappings in the Consistency Group must be synchronized correctly with respect to I/Os that are directed at the volumes by using the **startfcmap** or **startfcconsistgrp** command. The following actions occur during the execution of the **startfcmap** command or the **startfcconsistgrp** command: ► New reads and writes to all source volumes in the Consistency Group are paused in the cache layer until all ongoing reads and writes beneath the cache layer are completed. ► After all FlashCopy mappings in the Consistency Group are paused, the internal cluster state is set to allow FlashCopy operations. ► After the cluster state is set for all FlashCopy mappings in the Consistency Group, read and write operations continue on the source volumes. ► The target volumes are brought online. As part of the **startfcmap** or **startfcconsistgrp** command, read and write caching is enabled for both the source and target volumes.
Modify	You can modify the following FlashCopy mapping properties: ► FlashCopy mapping name ► Clean rate ► Consistency Group ► Copy rate (for background copy) ► Automatic deletion of the mapping when the background copy is complete
Stop	There are two separate mechanisms by which a FlashCopy mapping can be stopped: ► You have issued a command. ► An I/O error has occurred.
Delete	This command requests that the specified FlashCopy mapping be deleted. If the FlashCopy mapping is in the Stopped state, the **force** flag must be used.
Flush failed	If the flush of data from the cache cannot be completed, the FlashCopy mapping enters the Stopped state.
Copy complete	After all of the source data has been copied to the target and there are no dependent mappings, the state is set to Copied. If the option to automatically delete the mapping after the background copy completes is specified, the FlashCopy mapping is deleted automatically. If this option is not specified, the FlashCopy mapping is not deleted automatically and can be reactivated by preparing and starting again.
Bitmap online/offline	The node has failed.

5.4.1 FlashCopy mapping states

This section describes the states of a FlashCopy mapping in more detail. This concept is applicable for FlashCopy Mappings and FlashCopy Consistency Groups.

Idle_or_copied

Read and write caching is enabled for both the source and the target. A FlashCopy mapping exists between the source and target, but the source and target behave as independent volumes in this state.

Copying

The FlashCopy indirection layer governs all I/O to the source and target volumes while the background copy is running. The background copy process is copying "grains" from the source to the target.

Reads and writes are executed on the target as though the contents of the source were instantaneously copied to the target during the **`startfcmap`** or **`startfcconsistgrp`** command. The source and target can be independently updated. Internally, the target depends on the source for certain tracks. Read and write caching is enabled on the source and the target.

Stopped

The FlashCopy was stopped either by a user command or by an I/O error.

When a FlashCopy mapping is stopped, the integrity of the data on the target volume is lost. Therefore, while the FlashCopy mapping is in this state, the target volume is in the Offline state. To regain access to the target, the mapping must be started again (the previous point-in-time will be lost) or the FlashCopy mapping must be deleted. The source volume is accessible, and read and write caching is enabled for the source. In the Stopped state, a mapping can either be prepared again or deleted.

Stopping

The mapping is in the process of transferring data to a dependent mapping. The behavior of the target volume depends on whether the background copy process had completed while the mapping was in the Copying state. If the copy process had completed, the target volume remains online while the stopping copy process completes. If the copy process had not completed, data in the cache is discarded for the target volume. The target volume is taken offline, and the stopping copy process runs. After the data has been copied, a stop complete asynchronous event notification is issued. The mapping will move to the Idle/Copied state if the background copy has completed or to the Stopped state if the background copy has not completed. The source volume remains accessible for I/O.

Suspended

The FlashCopy was in the Copying or Stopping state when access to the metadata was lost. As a result, both the source and target volumes are offline and the background copy process has been halted.

When the metadata becomes available again, the FlashCopy mapping will return to the Copying or Stopping state. Access to the source and target volumes will be restored, and the background copy or stopping process will resume. Unflushed data that was written to the source or target before the FlashCopy was suspended is pinned in cache until the FlashCopy mapping leaves the Suspended state.

Preparing

The FlashCopy is in the process of preparing the mapping. While in this state, data from cache is destaged to disk and a consistent copy of the source exists on disk. At this time, cache is operating in write-through mode and therefore writes to the source volume will experience additional latency. The target volume is reported as online, but it will not perform reads or writes. These reads and writes are failed by the SCSI front end.

Before starting the FlashCopy mapping, it is important that any cache at the host level, for example, buffers on the host operating system or application, are also instructed to flush any outstanding writes to the source volume.

Performing the cache flush that is required as part of the **startfcmap** or **startfcconsistgrp** command causes I/Os to be delayed waiting for the cache flush to complete. To overcome this problem, FlashCopy supports the **prestartfcmap** or **prestartfcconsistgrp** commands, which prepare for a FlashCopy start while still allowing I/Os to continue to the source volume.

In the Preparing state, the FlashCopy mapping is prepared by the following steps:

1. Flushing any modified write data associated with the source volume from the cache. Read data for the source will be left in the cache.
2. Placing the cache for the source volume into write-through mode, so that subsequent writes wait until data has been written to disk before completing the **write** command that is received from the host.
3. Discarding any read or write data that is associated with the target volume from the cache.

Prepared

While in the Prepared state, the FlashCopy mapping is ready to perform a start. While the FlashCopy mapping is in this state, the target volume is in the Offline state. In the Prepared state, writes to the source volume experience additional latency, because the cache is operating in write-through mode.

Summary of FlashCopy mapping states

Table 5-6 lists the various FlashCopy mapping states and the corresponding states of the source and target volumes.

Table 5-6 FlashCopy mapping state summary

State	Source		Target	
	Online/Offline	**Cache state**	**Online/Offline**	**Cache state**
Idling/Copied	Online	Write-back	Online	Write-back
Copying	Online	Write-back	Online	Write-back
Stopped	Online	Write-back	Offline	N/A
Stopping	Online	Write-back	Online if copy complete Offline if copy not complete	N/A
Suspended	Offline	Write-back	Offline	N/A
Preparing	Online	Write-through	Online but not accessible	N/A
Prepared	Online	Write-through	Online but not accessible	N/A

Figure 5-10 is the FlashCopy mapping state diagram. It illustrates the states in which a mapping can exist, and which events are responsible for a state change.

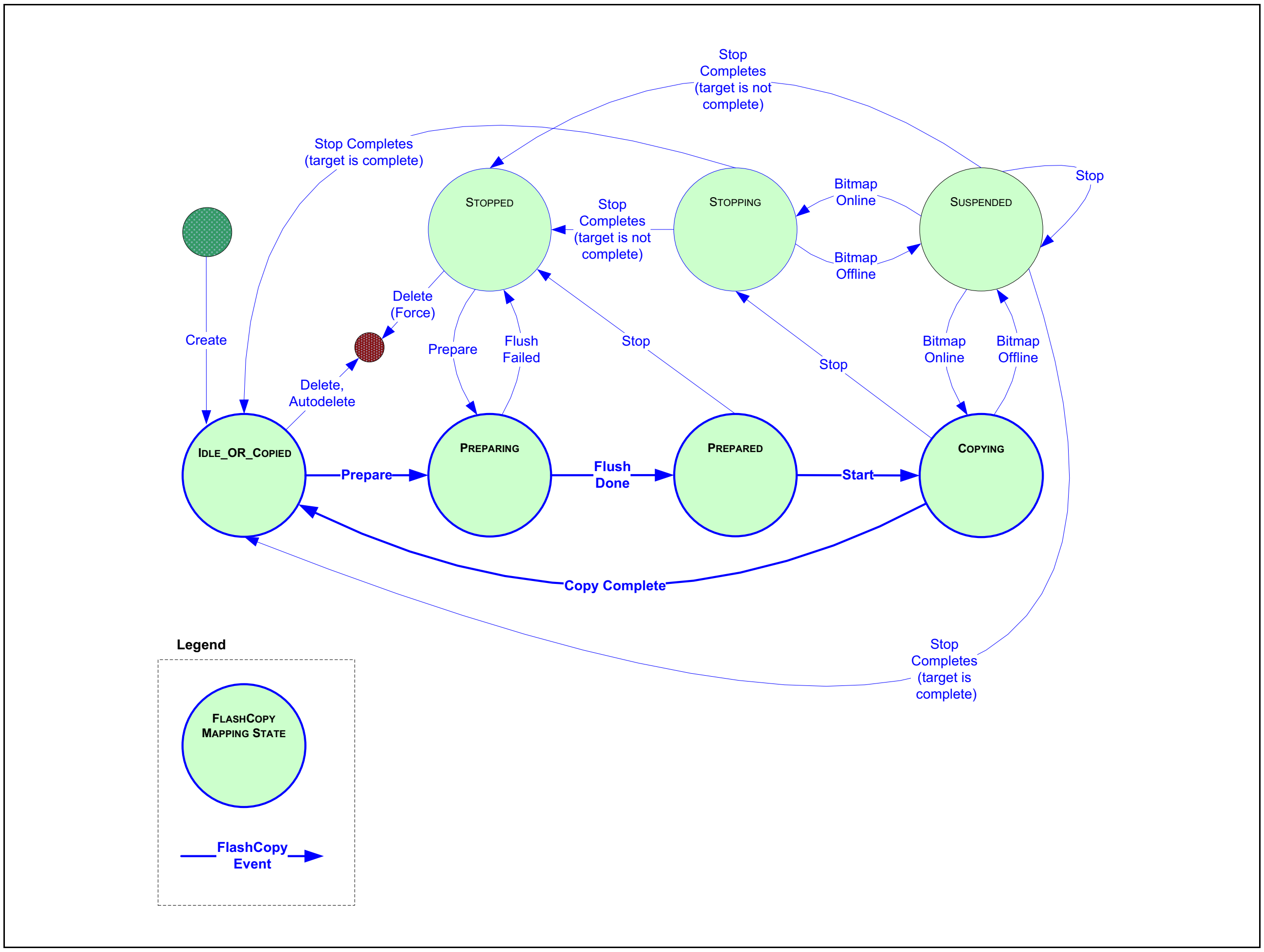


Figure 5-10 FlashCopy mapping states diagram

5.5 Dependencies between FlashCopy mappings

FlashCopy mappings, which are members in a tree, can depend on each other. A *dependency* is when one or more mappings are responsible for providing grains for another mapping. In this case, actions such as the deletion of one mapping can affect other mappings that depend on the mapping that is being deleted.

A dependency between mappings exists only as long as a source volume of a mapping does not physically hold all grains. This can happen only when a tree contains preceding mappings that have not yet completed the background copy process to establish an independent clone of the source volume.

5.5.1 Linked lists of mappings and resulting dependencies

This section explains linked lists, dependencies, and the effect on mappings in more detail.

Linked lists

Linked lists are dynamic internal data structures that hold information about active FlashCopy mappings in an MTFC setup, a FlashCopy cascade, or a combination of both. A mapping is inserted into a linked list of mappings when it is being started, not when the mapping is created. You can set up a cycle of mappings, but not all of them can be active at the same time; no cycle is allowed in a linked list.

Dependencies and effect on mappings

Dependencies in a dependency chain exist as long as a single mapping contains unsplit grains. When a target volume in a mapping has received all of the grains from the source volume, the target volume is no longer dependent on the source volume. Similarly, in a linked list of mappings, when all of the grains from a target volume (acting as a source) have been copied to a dependent target volume, the dependency ceases to exist.

Dependencies between mappings have effects, such as the dependent mappings stop when a mapping on which they depend stops (either because it is force-stopped or has an error). Also, the target volume of the dependent mappings go offline if the target volume of the mapping on which they depend goes offline, which happens when the source volume goes offline while the background copy is incomplete.

Dependencies with Cascaded FlashCopy

In the case of CFC, the dependency exists because the source volume of a mapping can be the target volume of another mapping that does not hold all of the data physically (because the background copy process of the preceding mapping is not complete). Thus, grains can still reside on the source volume of the preceding mapping in the cascade. A cascaded mapping cannot be prepared in case active mappings exist further down the cascade.

For Cascaded Mappings, if mappings in the linked list below the current mapping are active (not Stopped or Idle Copied), then this mapping cannot be started. To start this mapping, the downstream mappings must be stopped. See more details about mapping states in 5.4.1, “FlashCopy mapping states” on page 173.

Dependencies with Multiple Target FlashCopy

With MTFC, there is no obvious “preceding” mapping because all target volumes share the same source volume, which holds all of the data physically. Nonetheless, because of the implementation of FlashCopy using linked lists an order has been established, which makes mappings precede other mappings. In the case of many mappings with the same source, to avoid too much load on the source volume, MTFC is implemented so that mappings try to get grains from preceding mappings in the list, which are newer members of the list (that is, mappings started later).

Position of mappings and resulting dependencies

The first mapping in a linked list simply maps one source volume to a target volume (first member, oldest mapping). New mappings are inserted into the linked list (when the mapping is being started) using the following rules:

- A new cascaded mapping is inserted behind the mapping whose target thin-provisioned volume is being used as the source volume.
- A new multiple target mapping is inserted in front of the next oldest mapping of the same source volume.
- If multiple mappings are started at the same time, because they are in the same Consistency Group, an unspecified ordering is chosen.

The mapping in the first position, at the “head” of the linked list (not necessarily the one being inserted first), does not depend on other mappings. Every other succeeding mapping depends on the preceding mappings in the list.

The illustrations in this section show the way mappings are inserted into a linked list, depending on their being set up as an MTFC, or a CFC, or as both.

A simple MTFC is shown in Figure 5-11.

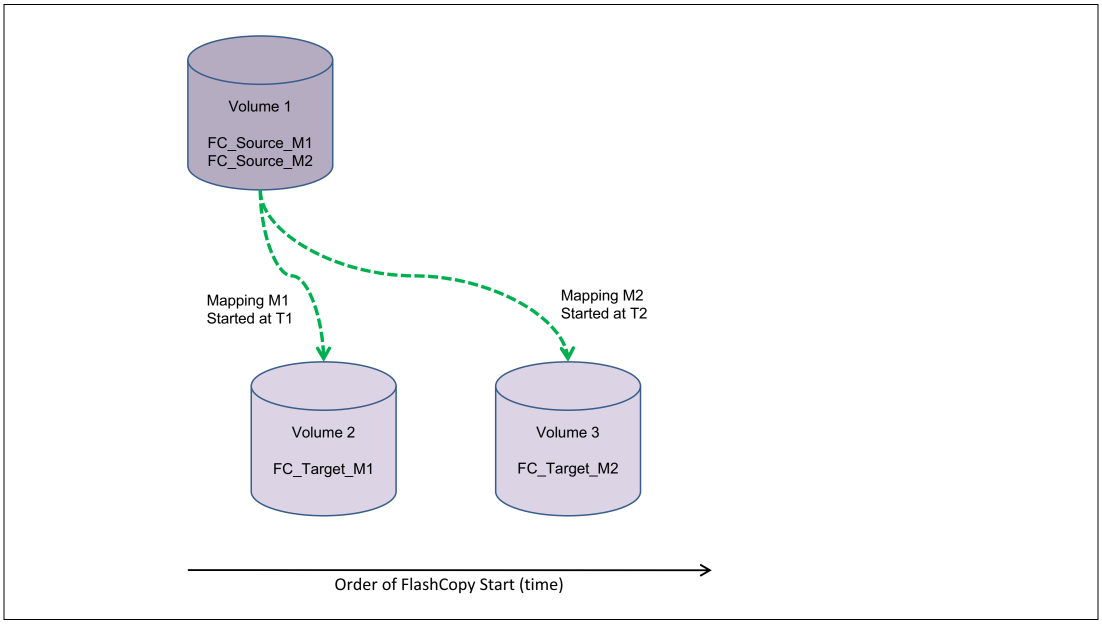


Figure 5-11 Multiple Target FlashCopy: m2 has started after m1

Both mappings, sharing the same source volume, result in an MTFC, which results in a linked list consisting of two mappings that is shown in Figure 5-12. These lists start with the leftmost mapping, which is called the *head* of the list.

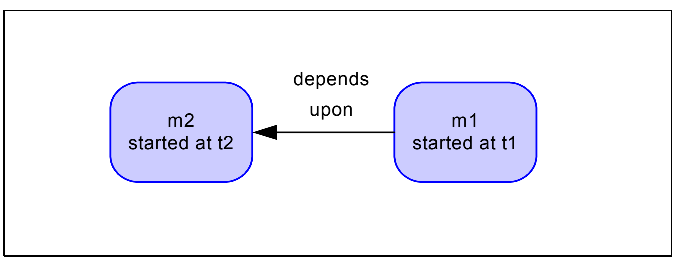


Figure 5-12 Linked list of MTFC mappings m1 and m2

The arrow in Figure 5-12 originates from the mapping that is dependent on the mapping to which the arrow points.

Mapping m2, which was started after mapping m1, precedes mapping m1 in the list. Mapping m1 now depends on mapping m2. This dependency means that for any grains that are not located in mapping m1, the target volume of mapping m2 will be used. Figure 5-13 shows which volume provide grains for volume 2.

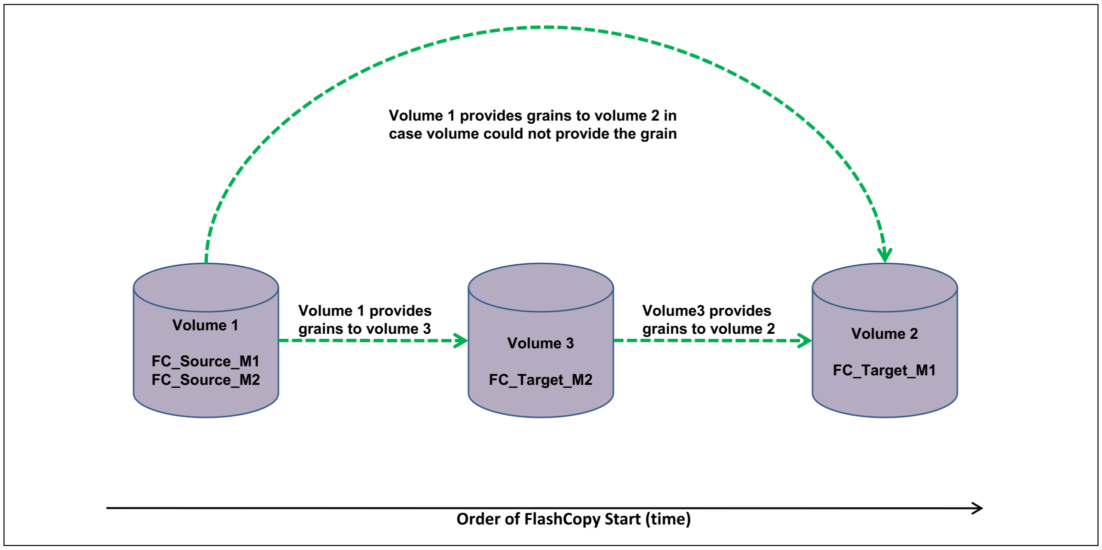


Figure 5-13 Volumes provide grains

A simple CFC mapping is shown in Figure 5-14.

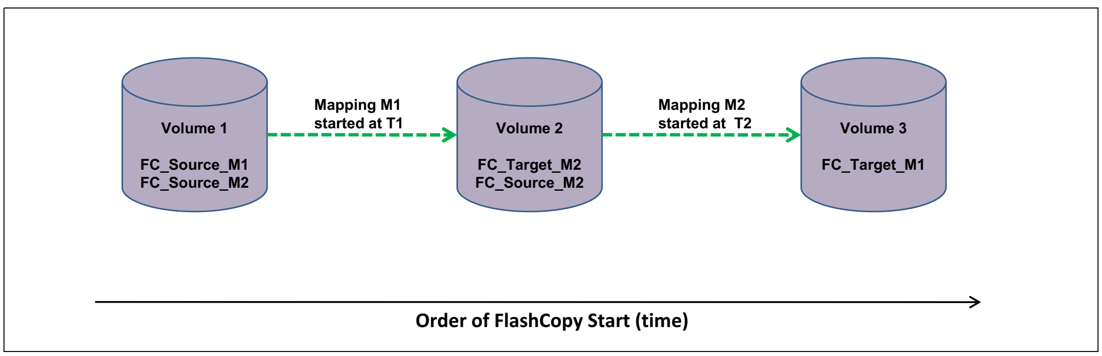


Figure 5-14 Cascaded FlashCopy (CFC): m2 started after m1

Figure 5-15 on page 179 shows how the volumes provide grains.

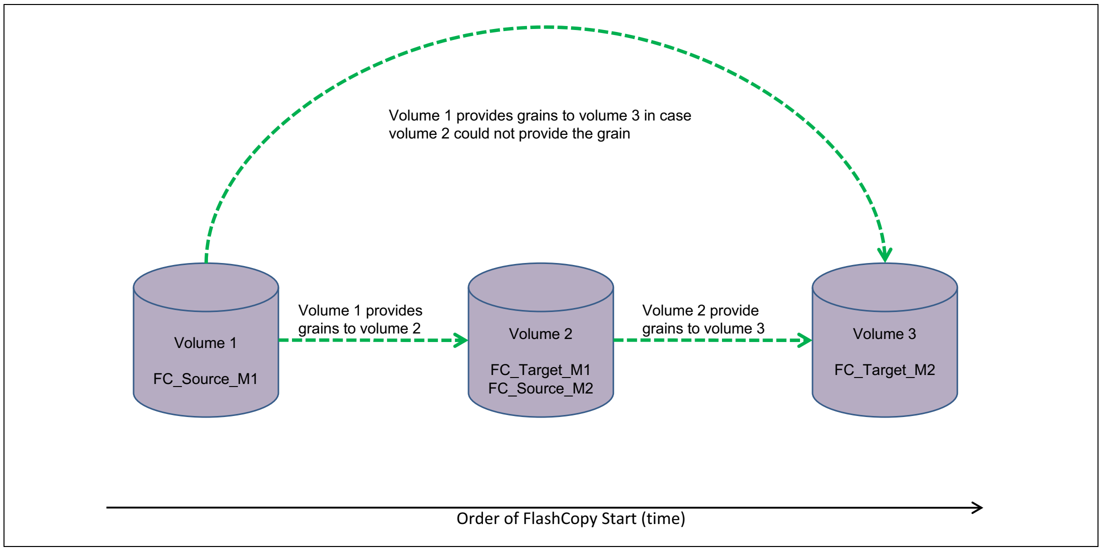


Figure 5-15 Volumes providing grains

The rules for inserting mappings in the list also apply when combining MTFC and CFC. An example of this type of a tree is shown in Figure 5-16.

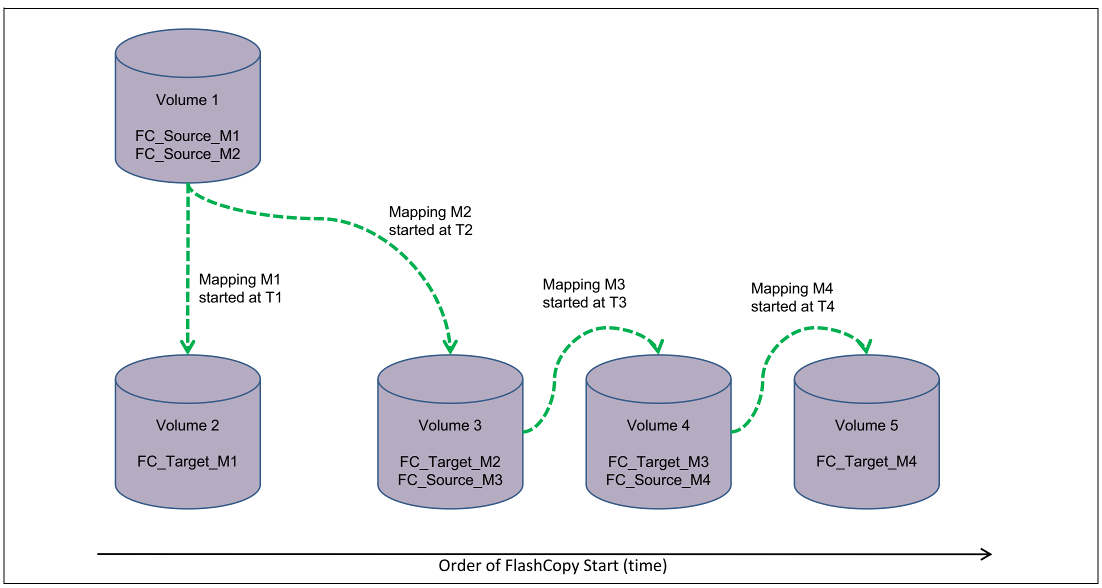


Figure 5-16 Combined MTFC and CFC

The resulting linked list is shown in Figure 5-17.

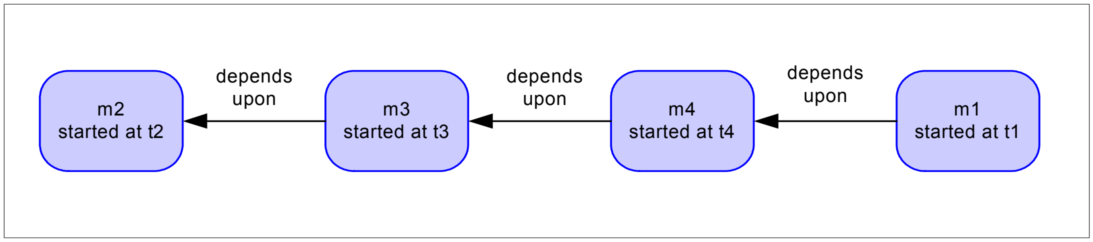


Figure 5-17   Linked list of combined MTFC and CFC

Figure 5-18 shows Cascaded FlashCopy.

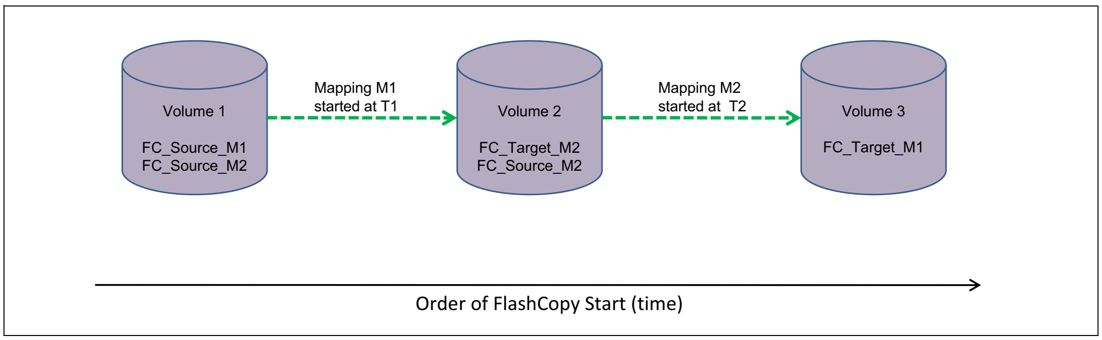


Figure 5-18   Cascaded FlashCopy

Writes to target volumes in a linked list

For MTFC, writes to a target volume cause writes to target volumes of the next dependent mapping. When an application writes to a target volume, the target volumes content changes compared to the content of the source volume for that specific point in time.

This change, however, will not happen on other mappings. To retain the data of the point-in-time copy for other dependent mappings, the target volume of the preceding mapping copies the grains to be changed to the target volume of the dependent mapping. For CFC, a write to a target volume causes a write to the target volume of the next mapping in the cascade. This write to an unsplit grain on this target volume merges the data from the target volume of the previous mapping in the cascade with the storage client write data and writes the result to the next mapping in the cascade.

5.5.2 Stopping FlashCopy mappings within a tree

When a FlashCopy mapping in a tree must be stopped and deleted, you have to work with dependencies. Stopping a FlashCopy mapping has an impact on dependent FlashCopy mappings for which a volume in the stopped FlashCopy mapping holds grains that were still not split (unique within the tree). To prevent that impact from happening, the FlashCopy mapping enters the Stopping state in which state the unique grains are copied from the (soon to be stopped) FlashCopy mapping.

When stopping a FlashCopy mapping where the target volume is already an independent copy, the target volume stays online while the FlashCopy mapping is in the Stopping state. If the target volume is *not* an independent copy, it goes offline while entering the Stopping state. Therefore, the target volume is not accessible to storage clients for what might be an unacceptably long time. This situation is handled by the Cleaning mode.

Stopping state of a FlashCopy mapping in a FlashCopy tree

To avoid impacting FlashCopy mappings that depend on the FlashCopy mapping to be stopped, the Stopping state is entered before it goes into the Stopped state.

The background copy process for a FlashCopy mapping (in case one was active) will stop after the FlashCopy mapping enters the Stopping state. Instead of the background copy process, a stopping copy process gets activated. It finds all of the grains on the FlashCopy mapping to be stopped that are unique in the tree and copies these grains to the next mapping in the linked list that is in the copying state (which means it is not in the Idle_Or_Copied state and is fully copied). When this process of duplicating the unique grains is complete, the FlashCopy mapping to be stopped enters the Stopped state (and might be deleted).

Stopping copy process copy rate

The stopping copy process will run at the same rate as the user specified for the background copy process (see more details about background copy in 5.2.9, "Background copy" on page 161). For FlashCopy mappings with no background copy (the background copy rate is set to 0), the stopping copy process will use a copy rate of 50 (the same as the default background copy rate).

Cleaning mode of a FlashCopy mapping in a FlashCopy tree

To resolve the situation where a target volume is offline because the FlashCopy mapping to which it belongs has entered the Stopping state, an additional method of copying grains has been implemented, which is the *Cleaning mode*. In this mode, grains are copied from the FlashCopy mapping to be stopped to a dependent FlashCopy mapping while it is still in the Copying state (and while its target volume is still accessible to the storage client). Information about the progress is contained in a new Cleaning Progress field introduced in the query output of the mapping.

After the Cleaning is complete (all unique grains are copied), the FlashCopy mapping can be stopped, and it will be in the Stopped state immediately. If the FlashCopy mapping is stopped before the cleaning process completed, it enters the Stopping state first, and the remaining grains are copied. After this copy completes, the FlashCopy mapping enters the Stopped state.

Cleaning mode copy rate

The Cleaning mode is implemented using a Cleaning Rate parameter. The cleaning rate for a FlashCopy mapping can be set independently of the background copy rate with a parameter when creating or modifying a FlashCopy mapping.

The copy rates that are used for the Cleaning mode in different states of FlashCopy mappings are:

- FlashCopy mapping in the state: Copying

  If a background copy rate for a FlashCopy mapping has been set (the background copy rate is not 0), and the FlashCopy mapping is not copy complete, the FlashCopy mapping will perform background copy at the set rate.

  If a background copy rate for a FlashCopy mapping has been set (the background copy rate is not 0) and the FlashCopy is copy complete, the FlashCopy mapping will perform cleaning at the set cleaning rate.

  If background copy for a FlashCopy mapping has been disabled (the background copy rate is 0), the FlashCopy mapping will perform cleaning at the set cleaning rate.

- FlashCopy mapping in the state: Idle_Or_Copied

  If a background copy rate for a FlashCopy mapping has been set (the background copy rate is not 0), and the FlashCopy mapping *is* copy complete, the mapping performs cleaning at the cleaning rate.

  If no background copy rate for a FlashCopy mapping has been set (the background copy rate is 0), the FlashCopy mapping cleans the FlashCopy mapping at the cleaning rate.

- FlashCopy mapping in the state: Stopping

  The FlashCopy mapping cleans at the cleaning rate unless the cleaning rate is 0, in which case it cleans at a rate of 50.

5.6 Practical example of a mapping tree

The flexibility of FlashCopy lets you create test setups that can simplify your processes. For instance, application testing can benefit from using FlashCopy functions.

5.6.1 FlashCopy mapping setup

In our example, we build a FlashCopy mapping tree that includes CFC and MTFC. It also uses Incremental FlashCopy and SEFC. Figure 5-19 illustrates our setup.

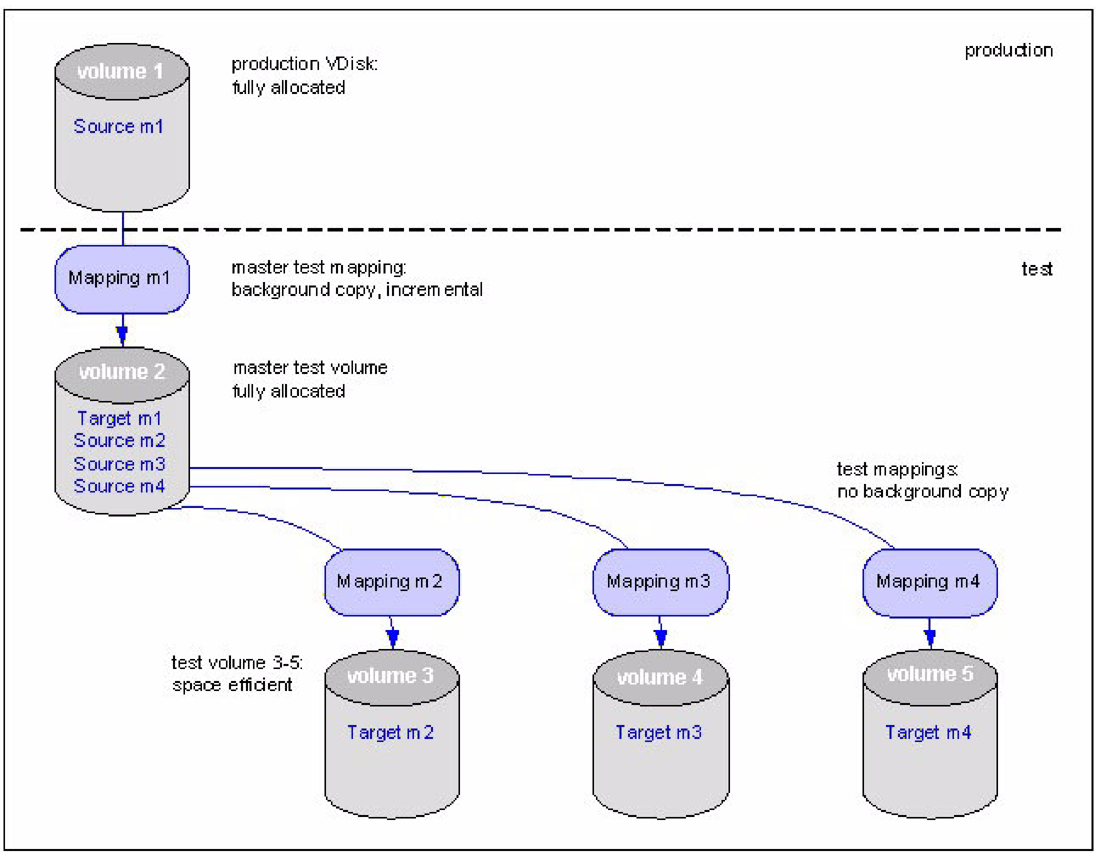


Figure 5-19 FlashCopy mapping tree example for application testing

The first mapping, the master test mapping `m1`, maps the production Volume 1 to the master test mapping target Volume 2. Both volumes in this mapping are fully allocated. The mapping m1 is set up to use background copy and to be incremental.

Volume 2 is the source volume for the test mappings `m2` - `m4`. This setup combines CFC with MTFC in this tree. The test mappings are all set up without background copy, and the target volumes for the test mappings are Space-Efficient.

5.6.2 Workflow to set up and run FlashCopy mappings

Follow these steps to set up the test mappings:

1. Create mapping m1 between the production Volume 1 and the master test Volume 2:
   - Both volumes are FA.
   - Mapping is with background copy and incremental.
2. Create mappings m2 - m4 between the master test Volume 2 and the test Volumes 3-5:
   - Source volume is FA.
   - Target volumes are SE.
   - Mapping is without using background copy and cleaning.

Follow these steps to use the mappings for the first time (create the FlashCopy mapping tree):

1. Start master test mapping m1.
2. Wait for mapping m1 to finish its background copy process.

   Mapping enters Idle_or_Copied state.
3. Start test mappings in the desired order.
4. Test the applications on the target volumes of the test mappings.

Follow these steps to refresh the master test mapping target volume and rerun tests on test volumes:

1. Finish the application testing and unmount test volumes.
2. Stop all of the test mappings.
3. Restart master test mapping m1.

   Mapping m1 target volume gets refreshed incrementally.
4. Wait for mapping m1 to finish its background copy process.
5. Restart test mappings in the desired order.
6. Retest applications on the target volumes of the test mappings.

5.6.3 Characteristics of this setup and notes about the steps

Regarding our examples, note the following characteristics about the setup:

- The production volume is Fully Allocated (FA), as is the master test volume.

  There is no benefit to having the master test volume Space-Efficient (SE), because we are going to use background copy.

- The master test mapping is set up to use background copy.

  This way, when the background copy process has finished, a clone of the production volume has been created. Any test FlashCopy mapping that uses the master test volume as a source does not put additional workload on the production volume.

- We use the incremental flag for the master test mapping.

  This way it can be restarted to create an actualized master test volume in less time and without affecting the SVC and the production volume too much.

- The test mappings are all set up without background copy, and the target volumes for the test mappings are Space-Efficient.

  This design saves space, because without background copy, only regions where data changes on the source volume are copied, and only writes to the target volume take space. Because the source volume for the mappings is used as the master test volume where we do not change data, only writes to the target can cause more space to be allocated on the target volumes of the test mappings.

- We wait for the master test mapping to finish its background copy process.

  If we start any of the test mappings before reads to the test mappings where the data is not yet copied to the target volume of the master test mapping (the source for the test mappings), then we will be redirected to the production volume. This redirection puts additional workload on the production volume, which we want to avoid.

- We stop the test mappings before we refresh the master test mapping target volume.

  These target volumes are Space-Efficient. Because the mappings depend on the master test mapping, restarting the master test mapping causes the data of the target volume of the master test mapping to be copied (cleaned) to the next dependent mapping in the linked list (the most recently started). This action then makes this SE test target volume unnecessarily large (fully allocated). If we restart the master test mapping before we start every single test mapping, every target volume of the test mappings is fully allocated.

5.6.4 Workflow to set up and run FlashCopy mappings

Follow these steps to set up the test mappings:

1. Create mapping m1 between the production volume 1 and the test volume 2:
   - Both volumes are Fully Allocated (FA); that is, it is not a thin-provisioned volume.
   - Mapping is with background copy (non NOCOPY) and incremental. For more details, see 5.2.9, “Background copy” on page 161 and 5.2.7, “Incremental FlashCopy” on page 158.
2. Create mappings m2 - m4 between the master test volume 2 and the test volume 3 - 5:
   - Source volume is FA
   - Target volumes are thin-provisioned
   - Mapping is without using background copy and cleaning.

Follow these steps to use the mappings for the first time (create the FlashCopy mapping tree):

1. Start master test mapping m1.
2. Wait for mapping m1 to finish its background copy process.

   Mapping enters Idle_or_Copied state.
3. Start test mappings in the desired order.
4. Test the applications on the target volumes of the test mappings.

Follow these steps to refresh the master test mapping target volume and rerun tests on test volumes:

1. Finish the application testing and unmount test volumes.
2. Stop all of the test mappings.
3. Restart test mapping m1.

   Mapping m1 target volume gets refreshed incrementally.
4. Wait for mapping m1 to finish its background copy process.
5. Restart test mappings in the desired order.
6. Retest applications on the target volumes of the test mappings.

5.6.5 Characteristics of this setup and notes about the steps

Regarding our examples, note the following characteristics about the setup:

- The production volume is Fully Allocated, as is the master test volume.

  There is no benefit to having the master test volume thin-provisioned because we are going to use background copy.
- The master test mapping is set up to use background copy.

  In this way, when the background copy process has finished, a clone of the production volume will have been created. Any test FlashCopy mapping that uses the master test volume as a source does not put additional workload on the production volume.
- We use the **`-incremental`** flag for the master test mapping.

  It can be restarted to create an updated point-in-time of test volume in less time and without affecting the cluster and the production volume.
- The test mappings are set up without background copy, and the target volumes for the test mappings are thin-provisioned.

  This design saves space, because without a background copy, only regions where data changes on the source volume are copied, and only writes to the target volume take space. Because the source volume for the mappings is used as the master test volume where we do not change data, only writes to the target can cause more space to be allocated on the target volumes of the test mappings.
- We wait for the master test mapping to finish its background copy process.

  If we start any of the test mappings *before* reads to the test mappings where the data is not yet copied to the target volume of the master test mapping (the source for the test mappings), we will be redirected to the production volume.
- We stop the test mappings before we refresh the master test mapping target volume.

  These target volumes are thin-provisioned. Because the mappings depend on the master test mapping, restarting the master test mapping causes the data of the target volume of the master test mapping to be copied (cleaned) to the next dependent mapping in the linked list (the most recently started). This action then makes this thin-provisioned test target volume unnecessarily large (fully allocated). If we restart the master test mapping before we start every single test mapping, every target volume of the test mappings is fully allocated.

6

Implementing FlashCopy

Through the use of examples and scenarios, in this chapter we explain and illustrate how to implement FlashCopy by using the command-line interface (CLI) and by using the graphical user interface (GUI). It is assumed that you have a fully functional cluster (SAN Volume Controller or Storwize V7000) environment.

The following topics are addressed:

- Copy considerations
- FlashCopy - CLI operation
- Example 1: FlashCopy using the CLI
- FlashCopy - GUI operation

6.1 Copy considerations

Regardless of whether you use FlashCopy to make one target disk, or multiple target disks, it is important to consider the application and the operating system in your planning. Even though the cluster can make an exact image of a disk with FlashCopy at the point in time that you require, it is pointless if the operating system, or more importantly, the application, cannot use the copied disk. Check with your application vendor if you have any doubts about this ability in regard to your application.

Data stored to a disk from an application normally goes through these steps:

1. The application records the data using its defined application programming interface. Certain applications might first store their data in application memory before sending it to disk at a later time. Normally, subsequent reads of the block just being written will get the block in memory if it is still there.

2. The application sends the data to a file. The file system accepting the data might buffer it in memory for a period of time.

3. The file system will send the I/O to a disk controller after a defined period of time (or even based on an event).

4. The disk controller might cache its write in memory before sending the data to the physical drive. In SAN Volume Controller or Storwize cluster, it will store the write in internal cache before sending I/O to the real disk controller.

5. The data is stored on the drive.

6.1.1 Command syntax and online help

Command prefix changes: The command prefixes are no longer needed when issuing a command. If you have existing scripts that use those prefixes, they will continue to function. You do not need to change your scripts.

Two major command sets are available:

- The command set allows you to query the various components within the SVC environment.
- The command set allows you to make changes to the various components within the SVC.

When the command syntax is shown, you will see certain parameters in square brackets, for example `[parameter]`. These brackets indicate that the parameter is optional in most if not all instances. Any information that is not in square brackets is required information. You can view the syntax of a command by entering one of the following commands:

`-?`	Shows a complete list of informational commands
`-?`	Shows a complete list of task commands
`commandname -?`	Shows the syntax of informational commands
`commandname -?`	Shows the syntax of task commands
`commandname -filtervalue?`	Shows the filters that you can use to reduce the output of the informational commands

Help: You can also use **`-h`** instead of **`-?`**, for example, the **`-h`** or **`commandname -h`** command.

If you look at the syntax of the command by typing **command name -?**, you often see **-filter** listed as a parameter. Be aware that the *correct* parameter is **-filtervalue**.

> **Tip:** You can use the up and down arrow keys on your keyboard to recall commands that were recently issued. Then, you can use the left and right, Backspace, and Delete keys to edit commands before you resubmit them.

Using the shortcuts command

You can use the **shortcuts** command to display a list of display or execution commands. This command produces an alphabetical list of actions that are supported. The ***command*** parameter must be for display commands or for execution commands. The ***model*** parameter allows for different shortcuts on different platforms: 2145 or 2076.

```
<command> Shortcuts <model>
```

See Example 6-1 (lines have been removed from the command output for brevity).

Example 6-1 The shortcuts command

```
IBM_2145:ITSO_SVC1:admin>shortcuts 2145
addcontrolenclosure
addhostiogrp
addhostport
addmdisk
addnode
addvdiskcopy
applydrivesoftware
applysoftware
cancellivedump
cfgportip
chhost
chiogrp
chldap
chldapserver
chlicense
chmdisk
chmdiskgrp
chnode
chnodehw
chpartnership
chquorum
chrcconsistgrp
mkemailserver
mkemailuser
mkfcconsistgrp
mkfcmap
mkhost
mkldapserver
mkmdiskgrp
mkpartnership
mkrcconsistgrp
mkrcrelationship
mksnmpserver
mksyslogserver
mkuser
```

```
mkusergrp
mkvdisk
mkvdiskhostmap
prmmdisk
rmmdiskgrp
rmnode
rmpartnership
rmportip
rmrcconsistgrp
triggerlivedump
writesernum
```

Using reverse-i-search

If you work on your cluster with the same PuTTy session for many hours and enter many commands, scrolling back to find your previous or similar commands can be a time-intensive task. In this case, using the **reverse-i-search** command can help you quickly and easily find any command that you have already issued in the history of your commands by using the **Ctrl+R** keys. **Ctrl+R** will allow you to interactively search through the command history as you type commands. Pressing **Ctrl+R** at an empty command prompt will give you a prompt, as shown in Example 6-2.

Example 6-2 Using reverse-i-search

```
IBM_2145:ITSO_SVC1:admin>lsiogrp
id name            node_count vdisk_count host_count
0  io_grp0         2          10          8
1  io_grp1         2          10          8
2  io_grp2         0          0           0
3  io_grp3         0          0           0
4  recovery_io_grp 0          0           0
(reverse-i-search)`i': lsiogrp
```

As shown in Example 6-2, we had executed an **lsiogrp** command. By pressing **Ctrl+R** and typing **sv**, the command that we needed was recalled from history.

6.1.2 Interoperation with Metro Mirror and Global Mirror

To provide better protection of the data, the FlashCopy can work together with Metro Mirror and Global Mirror. For example, we can perform a Metro Mirror copy to duplicate data from Site_1 to Site_2 and, then perform a daily FlashCopy operation to back up data to another location.

Table 6-1 on page 191 lists which combinations of FlashCopy and Remote Copy (Metro Mirror and Global Mirror) are supported.

Remote copy: Remote copy refers to Metro Mirror and Global Mirror.

Table 6-1 FlashCopy and Remote Copy interaction

Component	Remote Copy primary site	Remote Copy secondary site
FlashCopy source	Supported	Supported **Latency:** When the FlashCopy relationship is in the Preparing and Prepared states, the cache at the remote copy secondary site operates in write-through mode. This process adds additional latency to the already latent remote copy relationship.
FlashCopy destination	This is a supported combination. It has three restrictions: 1) issuing a **`stop -force`** may cause the Remote Copy relationship to need to be fully re-synced. 2) Code level must be 6.2.x or higher. 3)IO Group must be the same.	This is a supported combination with the main restriction that the FlashCopy mapping cannot be copying, stopping, or suspended. Otherwise the restrictions are the same as at the Remote Copy primary site.

6.1.3 FlashCopy: simulated scenario

In this section, we use a simulated scenario to illustrate how to use commands with PuTTY to perform FlashCopy administration.

Scenario description

We use the following scenario in both the CLI section and the GUI section. In the scenario, we want to FlashCopy the following volumes:

DB_Source Database files
Log_Source Database log files
App_Source Application files

We create Consistency Groups to handle the FlashCopy of `DB_Source` and `Log_Source`, because data integrity must be kept on `DB_Source` and `Log_Source`.

In our scenario, the application files are independent of the database, so we create a single FlashCopy mapping for `App_Source`. We will make two FlashCopy targets for `DB_Source` and `Log_Source` and therefore, two Consistency Groups. Figure 6-1 on page 192 illustrates the scenario.

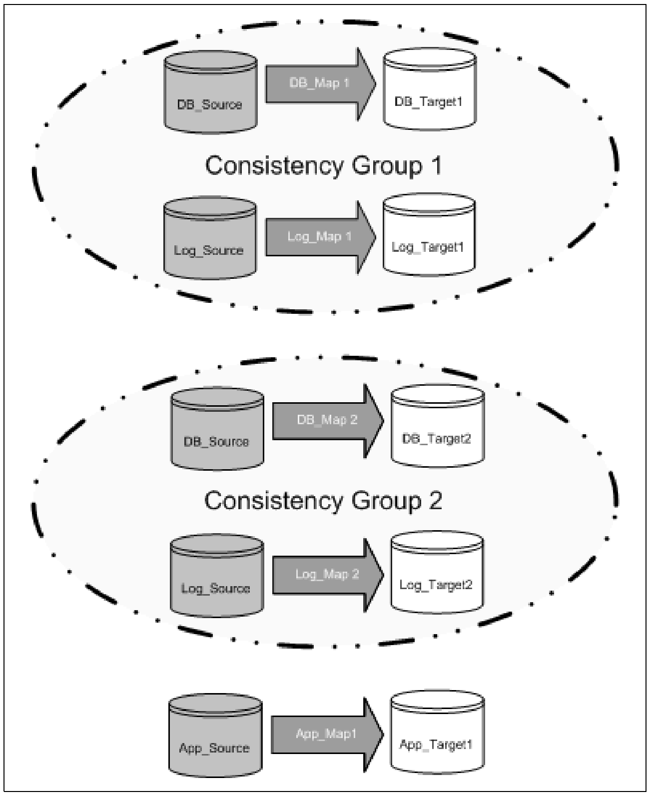


Figure 6-1 FlashCopy scenario

6.2 FlashCopy - CLI operation

This section describes FlashCopy and the CLI operation.

6.2.1 Setting up FlashCopy

In our case, we have already created the source and target volumes. The source and target volumes are identical in size, which is a requirement of the FlashCopy function:

- `DB_Source`, `DB_Target1`, and `DB_Target2`
- `Log_Source`, `Log_Target1`, and `Log_Target2`
- `App_Source` and `App_Target1`

To set up the FlashCopy, we perform the following steps:

1. Create two FlashCopy Consistency Groups:
   - `FCCG1`
   - `FCCG2`
2. Create FlashCopy mappings for Source volumes:
   - `DB_Source` FlashCopy to `DB_Target1`, the mapping name is `DB_Map1`
   - `DB_Source` FlashCopy to `DB_Target2`, the mapping name is `DB_Map2`
   - `Log_Source` FlashCopy to `Log_Target1`, the mapping name is `Log_Map1`
   - `Log_Source` FlashCopy to `Log_Target2`, the mapping name is `Log_Map2`
   - `App_Source` FlashCopy to `App_Target1`, the mapping name is `App_Map1`
   - Set Copyrate to 50

6.2.2 Creating a FlashCopy Consistency Group

To create a FlashCopy Consistency Group, we use the command **mkfcconsistgrp**. The ID of the new group is returned. If you have created several FlashCopy mappings for a group of volumes that contain elements of data for the same application, it might be convenient to assign these mappings to a single FlashCopy Consistency Group. Then you can issue a single **prepare** or **start** command for the whole FlashCopy Consistency Group, and in this case all files for a particular database are copied at the same time point in time.

In Example 6-3, the FCCG1 and FCCG2 Consistency Groups are created to hold the FlashCopy maps of DB and Log. This step is extremely important for FlashCopy on database applications, because it helps to maintain data integrity during FlashCopy.

Example 6-3 Creating two FlashCopy Consistency Groups

```
IBM_2145:ITSO_SVC2:admin>mkfcconsistgrp -name FCCG1
FlashCopy Consistency Group, id [1], successfully created

IBM_2145:ITSO_SVC2:admin>mkfcconsistgrp -name FCCG2
FlashCopy Consistency Group, id [2], successfully created
```

In Example 6-4, we checked the status of the Consistency Groups. Each Consistency Group has a status of empty.

Example 6-4 Checking the status

```
IBM_2145:ITSO_SVC2:admin>lsfcconsistgrp
id name  status
1  FCCG1 empty
2  FCCG2 empty
```

If you want to change the name of a Consistency Group, you can use the **chfcconsistgrp** command. Type **chfcconsistgrp -h** for help with this command.

6.2.3 Creating a FlashCopy mapping

To create a new FlashCopy mapping, use the **mkfcmap** command. This command maps a source volume to a target volume to prepare for subsequent copy.

When executed, this command creates a new FlashCopy mapping logical object. This mapping persists until it is deleted. The mapping specifies the source and destination volumes. The destination must be identical in size to the source or the mapping will fail. Issue the **lsvdisk -bytes** command to find the exact size of the source volume for which you want to create a target disk of the same size.

In a single mapping, source and destination cannot be on the same volume. A mapping is triggered at the point in time when the copy is required. The mapping can optionally be given a name and assigned to a Consistency Group. These groups of mappings can be triggered at the same time, thus enabling multiple volumes to be copied at the same time. This creates a consistent copy of multiple disks. A consistent copy of multiple disks is required for database products in which the database and log files reside on separate disks.

If any Consistency Group (ID or Name) is not specified, the mapping is assigned to the default group 0, which is a special group that cannot be started as a whole. Mappings in this group can only be started on an individual basis.

The *background copy rate* specifies the priority that must be given to completing the copy. If `0` is specified, the copy will not proceed in the background. The copy rate will determine how quickly the cluster will copy the data from the source to the target volume. Setting the copy rate to 0 (NOCOPY), the cluster will only copy blocks that have changed since the mapping was started on the source to the target volume (if the target volume is mounted, read write to a host is possible).The default is `50`.

> **Tip:** There is a parameter to delete FlashCopy mappings automatically after the completion of a background copy (volume in idle_or_copied state). Use the command:
>
> **`mkfcmap -autodelete`**
>
> Be aware that this command does *not* delete mappings in a cascade with dependent mappings, because it cannot get to the idle_or_copied state in this situation.

In Example 6-5, the first FlashCopy mapping for `DB_Source`, `Log_Source`, and `App_Source`is created.

Example 6-5 Create the first FlashCopy mapping for DB_Source, Log_Source, and App_Source

```
IBM_2145:ITSO_SVC2:admin>mkfcmap -source DB_Source -target DB_Target1 -name
DB_Map1 -consistgrp FCCG1
FlashCopy Mapping, id [0], successfully created

IBM_2145:ITSO_SVC2:admin>mkfcmap -source Log_Source -target Log_Target1 -name
Log_Map1 -consistgrp FCCG1
FlashCopy Mapping, id [1], successfully created

IBM_2145:ITSO_SVC2:admin>mkfcmap -source App_Source -target App_Target1 -name
App_Map1
FlashCopy Mapping, id [2], successfully created
```

Example 6-6 shows the command to create a second FlashCopy mapping for volume DB_Source and Log_Source.

Example 6-6 Create additional FlashCopy mappings

```
IBM_2145:ITSO_SVC2:admin>mkfcmap -source DB_Source -target DB_Target2 -name
DB_Map2 -consistgrp FCCG2
FlashCopy Mapping, id [3], successfully created

IBM_2145:ITSO_SVC2:admin>mkfcmap -source Log_Source -target Log_Target2 -name
Log_Map2 -consistgrp FCCG2
FlashCopy Mapping, id [4], successfully created
```

Example 6-7 shows the result of these FlashCopy mappings. The status of the mapping is idle_or_copied.

Example 6-7 Check the result of Multiple Target FlashCopy mappings

```
IBM_2145:ITSO_SVC2:admin>lsfcmap
id name     source_vdisk_id source_vdisk_name target_vdisk_id target_vdisk_name
group_id group_name status         progress copy_rate clean_progress incremental
partner_FC_id partner_FC_name restoring start_time rc_controlled
```

```
0  DB_Map1  3             DB_Source        4             DB_Target1       1
FCCG1    idle_or_copied 0      50      100         off
no                  no
1  Log_Map1 6             Log_Source       7             Log_Target1      1
FCCG1    idle_or_copied 0      50      100         off
no                  no
2  App_Map1 9           App_Source      10          App_Target1
idle_or_copied 0       50       100          off
no                  no
3  DB_Map2  3             DB_Source        5             DB_Target2       2
FCCG2    idle_or_copied 0      50      100         off
no                  no
4  Log_Map2 6             Log_Source       8             Log_Target2      2
FCCG2    idle_or_copied 0      50      100         off
no                  no
IBM_2145:ITSO_SVC2:admin>lsfcconsistgrp
id name  status
1  FCCG1 idle_or_copied
2  FCCG2 idle_or_copied
```

You can change the FlashCopy mapping attributes using the command **chfcmap**. Type **chfcmap -h** to see more details about how to use this command.

6.2.4 Preparing (pre-triggering) the FlashCopy mapping

At this point the mapping has been created, but the cache still accepts data for the source volumes (if the application is still being used). You can only trigger the mapping when the cache does not contain any data for FlashCopy source volumes. You must issue a **prestartfcmap** command to prepare a FlashCopy mapping to start. This command tells the cluster to flush the cache of any content for the source volume and to pass through any further write data for this volume.

When the **prestartfcmap** command is executed, the mapping enters the Preparing state. As soon as the preparation process is completed, the new state of the FlashCopy mapping is Prepared, and then mapping is ready for triggering.

Preparing and the subsequent triggering are usually performed on a Consistency Group concept. Only mappings belonging to Consistency Group `0` can be prepared on their own, because Consistency Group `0` is a special group that contains the FlashCopy mappings that do not belong to any Consistency Group.

In our scenario, `App_Map1` is not in a Consistency Group. In Example 6-8, we show how to initialize the preparation for `App_Map1`.

Another option is that you add the **-prep** parameter to the **startfcmap** command, which first prepares the mapping and then starts the FlashCopy.

In the example, we also show how to check the status of the current FlashCopy mapping. The status of App_Map1 is prepared.

Example 6-8   Prepare a FlashCopy without a Consistency Group

```
IBM_2145:ITSO_SVC2:admin>prestartfcmap App_Map1

IBM_2145:ITSO_SVC2:admin>lsfcmap App_Map1
id 2
```

```
name App_Map1
source_vdisk_id 9
source_vdisk_name App_Source
target_vdisk_id 10
target_vdisk_name App_Target1
group_id
group_name
status prepared
progress 0
copy_rate 50
start_time
dependent_mappings 0
autodelete off
clean_progress 0
clean_rate 50
incremental off
difference 0
grain_size 256
IO_group_id 0
IO_group_name io_grp0
partner_FC_id
partner_FC_name
restoring no
rc_controlled no
```

6.2.5 Preparing (pre-triggering) the FlashCopy Consistency Group

We use the **prestartfcconsistsgrp** command to prepare a FlashCopy Consistency Group. As with 6.2.4, "Preparing (pre-triggering) the FlashCopy mapping" on page 195, this command flushes the cache of any data that is destined for the source volume and forces the cache into the write-through mode until the mapping is started. The difference is that this command prepares a group of mappings (at a Consistency Group level) instead of one mapping.

When you have assigned several mappings to a FlashCopy Consistency Group, you only have to issue a single **prepare** command for the whole group to prepare all of the mappings at one time.

Example 6-9 shows how we prepare the Consistency Groups for DB and Log and check the result. After the command has executed all of the FlashCopy maps that we have, all of them are in Prepared status and all the Consistency Groups are in Prepared status as well. Now, we are ready to start the FlashCopy.

Example 6-9 Prepare a FlashCopy Consistency Group

```
IBM_2145:ITSO_SVC2:admin>prestartfcconsistgrp FCCG1
IBM_2145:ITSO_SVC2:admin>prestartfcconsistgrp FCCG2

IBM_2145:ITSO_SVC2:admin>lsfcconsistgrp FCCG1
id 1
name FCCG1
status prepared
autodelete off
FC_mapping_id 0
FC_mapping_name DB_Map1
```

```
FC_mapping_id 1
FC_mapping_name Log_Map1

IBM_2145:ITSO_SVC2:admin>lsfcconsistgrp
id name  status
1  FCCG1 prepared
2  FCCG2 prepared
```

6.2.6 Starting (triggering) FlashCopy mappings

The **startfcmap** command is used to start a single FlashCopy mapping. When invoked, a point-in-time copy of the source volume is created on the target volume.

When the FlashCopy mapping is triggered, it enters the Copying state. The way that the copy proceeds depends on the background copy rate attribute of the mapping. If the mapping is set to 0 (NOCOPY), only data that is subsequently updated on the source will be copied to the destination. We suggest that you use this scenario as a backup copy while the mapping exists in the Copying state. If the copy is stopped, the destination is unusable.

If you want to end up with a duplicate copy of the source at the destination, set the background copy rate greater than 0. This way, the system copies all of the data (even unchanged data) to the destination and eventually reaches the idle_or_copied state. After this data is copied, you can delete the mapping and have a usable point-in-time copy of the source at the destination.

In Example 6-10, after the FlashCopy is started, App_Map1 changes to copying status.

Example 6-10 Start App_Map1

```
IBM_2145:ITSO_SVC2:admin>startfcmap App_Map1
IBM_2145:ITSO_SVC2:admin>lsfcmap
id name     source_vdisk_id source_vdisk_name target_vdisk_id target_vdisk_name
group_id group_name status   progress copy_rate clean_progress incremental
partner_FC_id partner_FC_name restoring start_time   rc_controlled
0  DB_Map1  3             DB_Source        4             DB_Target1       1
FCCG1     prepared 0       50      0           off
no                     no
1  Log_Map1 6             Log_Source       7             Log_Target1      1
FCCG1     prepared 0       50      0           off
no                     no
2  App_Map1 9            App_Source      10          App_Target1
copying  0       50       100          off
no       110929113407 no
3  DB_Map2  3             DB_Source        5             DB_Target2       2
FCCG2     prepared 0       50      0           off
no                     no
4  Log_Map2 6             Log_Source       8             Log_Target2      2
FCCG2     prepared 0       50      0           off
no                     no
IBM_2145:ITSO_SVC2:admin>lsfcmap App_Map1
id 2
name App_Map1
source_vdisk_id 9
source_vdisk_name App_Source
target_vdisk_id 10
```

```
target_vdisk_name App_Target1
group_id
group_name
status copying
progress 0
copy_rate 50
start_time 110929113407
dependent_mappings 0
autodelete off
clean_progress 100
clean_rate 50
incremental off
difference 0
grain_size 256
IO_group_id 0
IO_group_name io_grp0
partner_FC_id
partner_FC_name
restoring no
rc_controlled no
```

6.2.7 Starting (triggering) FlashCopy Consistency Group

We execute the **startfcconsistgrp** command, as shown in Example 6-11, and afterward the database can be resumed. We have created two point-in-time consistent copies of the DB and Log volumes. After the execution, the Consistency Group and the FlashCopy maps are all in Copying status.

Example 6-11 Start FlashCopy Consistency Group

```
IBM_2145:ITSO_SVC2:admin>startfcconsistgrp FCCG1
IBM_2145:ITSO_SVC2:admin>startfcconsistgrp FCCG2
IBM_2145:ITSO_SVC2:admin>lsfcconsistgrp FCCG1
id 1
name FCCG1
status copying
autodelete off
FC_mapping_id 0
FC_mapping_name DB_Map1
FC_mapping_id 1
FC_mapping_name Log_Map1
IBM_2145:ITSO_SVC2:admin>
IBM_2145:ITSO_SVC2:admin>lsfcconsistgrp
id name  status
1  FCCG1 copying
2  FCCG2 copying
```

Tip: A status of copying but a progress indication of FlashCopy mapping is 100% means there are still dependencies on this target. As soon as there are no more dependencies, the status is changed to idle_copied.

6.2.8 Monitoring the FlashCopy progress

To monitor the background copy progress of the FlashCopy mappings, we issue the **lsfcmapprogress** command for each FlashCopy mapping.

Alternatively, you can also query the copy progress by using the **lsfcmap** command. As shown in Example 6-12, both DB_Map1 entries return information that the background copy is 23% completed, and Log_Map1 returns information that the background copy is 41% completed. DB_Map2 returns information that the background copy is 5% completed and Log_Map2 returns information that the background copy is 4% completed.

Example 6-12 Monitoring the background copy progress

```
IBM_2145:ITSO_SVC2:admin>lsfcmapprogress DB_Map1
id progress
0  23
IBM_2145:ITSO_SVC2:admin>lsfcmapprogress Log_Map1
id progress
1  41
IBM_2145:ITSO_SVC2:admin>lsfcmapprogress Log_Map2
id progress
4  4
IBM_2145:ITSO_SVC2:admin>lsfcmapprogress DB_Map2
id progress
3  5
IBM_2145:ITSO_SVC2:admin>lsfcmapprogress App_Map1
id progress
2  10
```

When the background copy has completed, the FlashCopy mapping enters the idle_or_copied state. When all FlashCopy mappings in a Consistency Group enter this status, the Consistency Group will be at idle_or_copied status.

When in this state, the FlashCopy mapping can be deleted and the target disk can be used independently if, for example, another target disk is to be used for the next FlashCopy of the particular source volume.

6.2.9 Stopping the FlashCopy mapping

The **stopfcmap** command is used to stop a FlashCopy mapping. This command allows you to stop an active (copying) or suspended mapping. When executed, this command stops a single FlashCopy mapping.

> **Tip:** In a Multiple Target FlashCopy environment, if you want to stop a mapping or group, consider whether you want to keep any of the dependent mappings. If not, issue the **stop** command with the **-force** parameter, which will stop all of the dependent maps and negate the need for the stopping copy process to run.

When a FlashCopy mapping is stopped, the target volume becomes invalid and is set offline by the cluster. The FlashCopy mapping needs to be prepared again or retriggered to bring the target volume online again.

Important: Only stop a FlashCopy mapping when the data on the target volume is not in use, or when you want to modify the FlashCopy mapping.

When a FlashCopy mapping is stopped, the target volume becomes invalid and is set offline by the cluster, if the mapping is in the Copying state and progress=`100`.

Example 6-13 shows how to stop the `App_Map1` FlashCopy. The status of `App_Map1` has changed to idle_or_copied.

Example 6-13 Stop App_Map1 FlashCopy

```
IBM_2145:ITSO_SVC2:admin>stopfcmap App_Map1

IBM_2145:ITSO_SVC2:admin>lsfcmap App_Map1
id 2
name App_Map1
source_vdisk_id 9
source_vdisk_name App_Source
target_vdisk_id 10
target_vdisk_name App_Target1
group_id
group_name
status idle_or_copied
progress 100
copy_rate 50
start_time 110929113407
dependent_mappings 0
autodelete off
clean_progress 100
clean_rate 50
incremental off
difference 100
grain_size 256
IO_group_id 0
IO_group_name io_grp0
partner_FC_id
partner_FC_name
restoring no
rc_controlled no
```

Stopping status: Without the **-force** flag being specified on **stopfcmap**, a status of Stopping indicates that flushing/cleaning to dependent targets is occurring. It will then go to the Stopped state.

6.2.10 Stopping the FlashCopy Consistency Group

The **stopfcconsistgrp** command is used to stop any active FlashCopy Consistency Group. It stops all mappings in a Consistency Group. When a FlashCopy Consistency Group is stopped for all mappings that are not 100% copied, the target volumes become invalid and are set offline by the cluster. The FlashCopy Consistency Group needs to be prepared again and restarted to bring the target volumes online again.

Important: Only stop a FlashCopy mapping when the data on the target volume is not in use, or when you want to modify the FlashCopy Consistency Group. When a Consistency Group is stopped, the target volume might become invalid and set offline by the cluster, depending on the state of the mapping.

As shown in Example 6-14, we stop the `FCCG1` and `FCCG2` Consistency Groups. The status of the two Consistency Groups has changed to `stopped`. Most of the FlashCopy mapping relationships now have the status of `stopped`. As shown, several have already completed the copy operation and are now in a status of `idle_or_copied`.

Example 6-14 Stop FCCG1 and FCCG2 Consistency Groups

```
IBM_2145:ITSO_SVC2:admin>stopfcconsistgrp FCCG1

IBM_2145:ITSO_SVC2:admin>stopfcconsistgrp FCCG2

IBM_2145:ITSO_SVC2:admin>lsfcconsistgrp
id name  status
1  FCCG1 idle_or_copied
2  FCCG2 idle_or_copied

IBM_2145:ITSO_SVC2:admin>lsfcmap -delim ,
id,name,source_vdisk_id,source_vdisk_name,target_vdisk_id,target_vdisk_name,group_
id,group_name,status,progress,copy_rate,clean_progress,incremental,partner_FC_id,p
artner_FC_name,restoring,start_time,rc_controlled
0,DB_Map1,3,DB_Source,4,DB_Target1,1,FCCG1,idle_or_copied,100,50,100,off,,,no,1109
29113806,no
1,Log_Map1,6,Log_Source,7,Log_Target1,1,FCCG1,idle_or_copied,100,50,100,off,,,no,1
10929113806,no
2,App_Map1,9,App_Source,10,App_Target1,,,idle_or_copied,100,50,100,off,,,no,110929
113407,no
3,DB_Map2,3,DB_Source,5,DB_Target2,2,FCCG2,idle_or_copied,100,50,100,off,,,no,1109
29113806,no
4,Log_Map2,6,Log_Source,8,Log_Target2,2,FCCG2,idle_or_copied,100,50,100,off,,,no,1
10929113806,no
```

Stopping status: Without the **`-force`** flag being specified on **`stopfcconsistgrp`**, a status of Stopping indicates flushing/cleaning to dependent targets is occurring. It will then go to the Stopped state.

6.2.11 Deleting the FlashCopy mapping

To delete a FlashCopy mapping, use the **`rmfcmap`** command. When the command is executed, it attempts to delete the specified FlashCopy mapping. If the FlashCopy mapping is stopped, the command fails unless the **`-force`** flag is specified. If the mapping is active (copying), it must first be stopped before it can be deleted.

Deleting a mapping only deletes the logical relationship between the two volumes. However, when issued on an active FlashCopy mapping using the **`-force`** flag, the delete renders the data on the FlashCopy mapping target volume as inconsistent.

> **Tip:** If you want to use the target volume as a normal volume, monitor the background copy progress until it is complete (100% copied). Then delete the FlashCopy mapping.
>
> Another option is to set the **-autodelete** option when creating the FlashCopy mapping.

As shown in Example 6-15, we delete `App_Map1`.

Example 6-15 Delete App_Map1

```
IBM_2145:ITSO_SVC2:admin>rmfcmap App_Map1
```

6.2.12 Deleting the FlashCopy Consistency Group

The **rmfcconsistgrp** command is used to delete a FlashCopy Consistency Group. When executed, this command deletes the specified Consistency Group. If there are mappings that are members of the group, the command fails unless the **-force** flag is specified.

If you want to delete all of the mappings in the Consistency Group as well, first delete the mappings and then delete the Consistency Group.

As shown in Example 6-16, we delete all maps and Consistency Groups and then check the result.

Example 6-16 Remove fcmaps and fcconsistgrp

```
IBM_2145:ITSO_SVC2:admin>rmfcmap DB_Map1

IBM_2145:ITSO_SVC2:admin>rmfcmap DB_Map2

IBM_2145:ITSO_SVC2:admin>rmfcmap Log_Map1

IBM_2145:ITSO_SVC2:admin>rmfcmap Log_Map2

IBM_2145:ITSO_SVC2:admin>rmfcconsistgrp FCCG1

IBM_2145:ITSO_SVC2:admin>rmfcconsistgrp FCCG2

IBM_2145:ITSO_SVC2:admin>lsfcconsistgrp

IBM_2145:ITSO_SVC2:admin>lsfcmap

IBM_2145:ITSO_SVC2:admin>
```

6.2.13 Migrating a volume to a thin-provisioned volume

Use the following scenario to migrate a volume to a thin-provisioned volume:

1. Create a thin-provisioned target volume with exactly the same size as the volume that you want to migrate.

   Example 6-17 on page 203 shows the details of a volume with ID `11`. It has been created as a thin-provisioned volume with the same size as the `App_Source` volume.

Example 6-17 Running the lsvdisk 11command

```
IBM_2145:ITSO_SVC2:admin>lsvdisk 11
id 11
name App_Source_SE
IO_group_id 0
IO_group_name io_grp0
status online
mdisk_grp_id 1
mdisk_grp_name Multi_Tier_Pool
capacity 10.00GB
type striped
formatted no
mdisk_id
mdisk_name
FC_id
FC_name
RC_id
RC_name
vdisk_UID 60050768018281BEE00000000000000B
throttling 0
preferred_node_id 1
fast_write_state empty
cache readwrite
udid
fc_map_count 0
sync_rate 50
copy_count 1
se_copy_count 1
filesystem
mirror_write_priority latency

copy_id 0
status online
sync yes
primary yes
mdisk_grp_id 1
mdisk_grp_name Multi_Tier_Pool
type striped
mdisk_id
mdisk_name
fast_write_state empty
used_capacity 0.41MB
real_capacity 221.17MB
free_capacity 220.77MB
overallocation 4629
autoexpand on
warning 80
grainsize 32
se_copy yes
easy_tier on
easy_tier_status active
tier generic_ssd
tier_capacity 0.00MB
tier generic_hdd
```

```
tier_capacity 221.17MB
```

2. Define a FlashCopy mapping in which the non-thin-provisioned volume is the source and the thin-provisioned volume is the target. Specify a copy rate as high as possible and activate the **-autodelete** option for the mapping (Example 6-18).

Example 6-18 Running the mkfcmap command

```
IBM_2145:ITSO_SVC2:admin>mkfcmap -source App_Source -target App_Source_SE -name
MigrtoThinProv -copyrate 100 -autodelete
FlashCopy Mapping, id [0], successfully created
IBM_2145:ITSO_SVC2:admin>lsfcmap 0
id 0
name MigrtoThinProv
source_vdisk_id 9
source_vdisk_name App_Source
target_vdisk_id 11
target_vdisk_name App_Source_SE
group_id
group_name
status idle_or_copied
progress 0
copy_rate 100
start_time
dependent_mappings 0
autodelete on
clean_progress 100
clean_rate 50
incremental off
difference 100
grain_size 256
IO_group_id 0
IO_group_name io_grp0
partner_FC_id
partner_FC_name
restoring no
rc_controlled no
```

3. Run the **prestartfcmap** command and the **lsfcmap MigrtoThinProv** command (Example 6-19).

Example 6-19 Running the prestartfcmap command

```
IBM_2145:ITSO_SVC2:admin>prestartfcmap MigrtoThinProv
IBM_2145:ITSO_SVC2:admin>lsfcmap MigrtoThinProv
id 0
name MigrtoThinProv
source_vdisk_id 9
source_vdisk_name App_Source
target_vdisk_id 11
target_vdisk_name App_Source_SE
group_id
group_name
status prepared
progress 0
copy_rate 100
```

```
start_time
dependent_mappings 0
autodelete on
clean_progress 0
clean_rate 50
incremental off
difference 100
grain_size 256
IO_group_id 0
IO_group_name io_grp0
partner_FC_id
partner_FC_name
restoring no
rc_controlled no
```

4. Run the **startfcmap** command (Example 6-20).

Example 6-20 Running the startfcmap command

```
IBM_2145:ITSO_SVC2:admin>startfcmap MigrtoThinProv
```

5. Monitor the copy process using the **lsfcmapprogress** comman (Example 6-21).

Example 6-21 Running the lsfcmapprogress command

```
IBM_2145:ITSO_SVC2:admin>lsfcmapprogress MigrtoThinProv
id progress
0  67
```

6. The FlashCopy mapping has been deleted automatically (Example 6-22).

Example 6-22 Running the lsfcmap command

```
IBM_2145:ITSO_SVC2:admin>lsfcmap MigrtoThinProv
id 0
name MigrtoThinProv
source_vdisk_id 9
source_vdisk_name App_Source
target_vdisk_id 11
target_vdisk_name App_Source_SE
group_id
group_name
status copying
progress 67
copy_rate 100
start_time 110929135848
dependent_mappings 0
autodelete on
clean_progress 100
clean_rate 50
incremental off
difference 100
grain_size 256
IO_group_id 0
IO_group_name io_grp0
partner_FC_id
partner_FC_name
```

```
restoring no
rc_controlled no
IBM_2145:ITSO_SVC2:admin>lsfcmapprogress MigrtoThinProv
CMMVC5804E The action failed because an object that was specified in the
command does not exist.
IBM_2145:ITSO_SVC2:admin>
```

An independent copy of the source volume (App_Source) has been created. The migration has completed, as shown in Example 6-23.

Example 6-23 lsvdisk App_Source

```
IBM_2145:ITSO_SVC2:admin>lsvdisk App_Source
id 9
name App_Source
IO_group_id 0
IO_group_name io_grp0
status online
mdisk_grp_id 1
mdisk_grp_name Multi_Tier_Pool
capacity 10.00GB
type striped
formatted no
mdisk_id
mdisk_name
FC_id
FC_name
RC_id
RC_name
vdisk_UID 60050768018281BEE000000000000009
throttling 0
preferred_node_id 1
fast_write_state empty
cache readwrite
udid
fc_map_count 0
sync_rate 50
copy_count 1
se_copy_count 0
filesystem
mirror_write_priority latency

copy_id 0
status online
sync yes
primary yes
mdisk_grp_id 1
mdisk_grp_name Multi_Tier_Pool
type striped
mdisk_id
mdisk_name
fast_write_state empty
used_capacity 10.00GB
real_capacity 10.00GB
free_capacity 0.00MB
overallocation 100
```

```
autoexpand
warning
grainsize
se_copy no
easy_tier on
easy_tier_status active
tier generic_ssd
tier_capacity 0.00MB
tier generic_hdd
tier_capacity 10.00GB
```

Real size: Independently of what you defined as the real size of the target thin-provisioned volume, the real size will be at *least* the capacity of the source volume.

To migrate a thin-provisioned volume to a fully allocated volume, you can follow the same scenario.

6.2.14 Reverse FlashCopy

You can also have a reverse FlashCopy mapping without having to remove the original FlashCopy mapping, and without restarting a FlashCopy mapping from the beginning.

In Example 6-24, `FCMAP_1` is the forward FlashCopy mapping, and `FCMAP_rev_1` is a reverse FlashCopy mapping. We have also a cascade `FCMAP_2` where its source is the `FCMAP_1` target volume, and its target is a separate volume named `Volume_FC_T1`.

In our example, after creating the environment, we started the `FCMAP_1` and later `FCMAP_2`.

As an example, we started `FCMAP_rev_1` without specifying the **`-restore`** parameter to show why we have to use it, and to show the message that is issued if you do not use it:

```
CMMVC6298E The command failed because a target VDisk has dependent FlashCopy
mappings.
```

When starting a reverse FlashCopy mapping, you must use the **`-restore`** option to indicate that you want to overwrite the data on the source disk of the forward mapping.

Example 6-24 Reverse FlashCopy

```
IBM_2145:ITSO_SVC2:admin>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 RC_change
3  Volume_FC_S    0           io_grp0       online 1            Multi_Tier_Pool
10.00GB  striped                     60050768018281BEE000000000000003 0
1         empty           0            0                      no
4  Volume_FC_T_S1 0           io_grp0       online 1            Multi_Tier_Pool
10.00GB  striped                     60050768018281BEE000000000000004 0
1         empty           0            0                      no
5  Volume_FC_T1   0           io_grp0       online 1            Multi_Tier_Pool
10.00GB  striped                     60050768018281BEE000000000000005 0
1         empty           0            0                      no

IBM_2145:ITSO_SVC2:admin>mkfcmap -source Volume_FC_S -target Volume_FC_T_S1 -name
FCMAP_1 -copyrate 50
```

```
FlashCopy Mapping, id [0], successfully created

IBM_2145:ITSO_SVC2:admin>mkfcmap -source Volume_FC_T_S1 -target Volume_FC_S -name
FCMAP_rev_1 -copyrate 50
FlashCopy Mapping, id [1], successfully created

IBM_2145:ITSO_SVC2:admin>mkfcmap -source Volume_FC_T_S1 -target Volume_FC_T1 -name
FCMAP_2 -copyrate 50
FlashCopy Mapping, id [2], successfully created

IBM_2145:ITSO_SVC2:admin>lsfcmap
id name       source_vdisk_id source_vdisk_name target_vdisk_id target_vdisk_name
group_id group_name status        progress copy_rate clean_progress incremental
partner_FC_id partner_FC_name restoring start_time rc_controlled
0  FCMAP_1    3          Volume_FC_S     4          Volume_FC_T_S1
idle_or_copied 0        50        100           off        1
FCMAP_rev_1     no                    no
1  FCMAP_rev_1 4          Volume_FC_T_S1   3          Volume_FC_S
idle_or_copied 0       50       100          off       0           FCMAP_1
no                  no
2  FCMAP_2    4          Volume_FC_T_S1   5          Volume_FC_T1
idle_or_copied 0       50       100          off
no                  no

IBM_2145:ITSO_SVC2:admin>startfcmap -prep FCMAP_1

IBM_2145:ITSO_SVC2:admin>startfcmap -prep FCMAP_2

IBM_2145:ITSO_SVC2:admin>lsfcmap
id name       source_vdisk_id source_vdisk_name target_vdisk_id target_vdisk_name
group_id group_name status        progress copy_rate clean_progress incremental
partner_FC_id partner_FC_name restoring start_time   rc_controlled
0  FCMAP_1     3              Volume_FC_S      4              Volume_FC_T_S1
copying 0       50        100           off        1            FCMAP_rev_1
no                    no
1  FCMAP_rev_1 4          Volume_FC_T_S1   3          Volume_FC_S
idle_or_copied 0       50       100          off       0           FCMAP_1
no                    no
2  FCMAP_2    4          Volume_FC_T_S1   5          Volume_FC_T1
copying      4       50       100          off
no       110929143739 no

IBM_2145:ITSO_SVC2:admin>startfcmap -prep FCMAP_rev_1
CMMVC6298E The command failed because a target VDisk has dependent FlashCopy
mappings.

IBM_2145:ITSO_SVC2:admin>startfcmap -prep -restore FCMAP_rev_1

IBM_2145:ITSO_SVC2:admin>lsfcmap
id name       source_vdisk_id source_vdisk_name target_vdisk_id target_vdisk_name
group_id group_name status  progress copy_rate clean_progress incremental
partner_FC_id partner_FC_name restoring start_time   rc_controlled
0  FCMAP_1    3          Volume_FC_S     4          Volume_FC_T_S1
copying 43       100       56            off        1            FCMAP_rev_1
no       110929151911 no
```

```
1  FCMAP_rev_1 4          Volume_FC_T_S1   3          Volume_FC_S
copying 56       100       43            off         0            FCMAP_1
yes       110929152030 no
2  FCMAP_2    4          Volume_FC_T_S1   5          Volume_FC_T1
copying 37       100       100           off
no        110929151926 no
```

As you can see in Example 6-24 on page 207, `FCMAP_rev_1` shows a restoring value of `yes` while the FlashCopy mapping is copying. After it has finished copying, the restoring value field will change to `no`.

6.2.15 Split-stopping FlashCopy maps

The **stopfcmap** command has a **-split** option. This option allows the source target of a map, which is 100% complete, to be removed from the head of a cascade when the map is stopped.

For example, if we have four volumes in a cascade (A → B → C → D), and the map A → B is 100% complete, using the **stopfcmap -split mapAB** command results in `mapAB` becoming idle_copied and the remaining cascade becoming B → C → D.

Without the **-split** option, volume A remains at the head of the cascade (A → C → D). Consider this sequence of steps:

1. A user takes a backup using the mapping A → B. A is the production volume. B is a backup.
2. At a later point, the user experiences corruption on A and so reverses the mapping to B → A.
3. The user then takes another backup from the production disk A, resulting in the cascade B → A → C.

Stopping A → B without the **-split** option results in the cascade B → C. Note that the backup disk B is now at the head of this cascade.

When the user next wants to take a backup to B, the user can still start mapping A → B (using the **-restore** flag), but the user cannot then reverse the mapping to A (B → A or C → A).

Stopping A → B with the **-split** option results in the cascade A → C. This action does not result in the same problem, because the production disk A is at the head of the cascade instead of the backup disk B.

6.3 Example: FlashCopy using the CLI

In this section, we implement FlashCopy using local volumes.

Our goal is to create a full consistent copy of production data for a given point in time at a remote location for disaster tolerance. We set up a Remote Copy Relationship to copy volumes to a second site using inter-cluster Metro Mirror. We want to take a consistent backup of our production data at the remote site.

We also want a full copy (a clone) of our production volumes taken on a daily basis and residing at the second location. This copy serves the purpose of providing a consistent point-in-time copy of the production data in case the data of both the primary and secondary of the Metro Mirror relationship become invalidated (for example, due to a logical error that affects both the primary and the secondary because they are in a synchronized copy relationship). This clone is a building block for our business continuity strategy.

Global Mirror consideration: Even though Global Mirror is “asynchronous,” this case also applies to Global Mirror because the data is copied over to the secondary volume as soon as possible.

Our solution is to set up FlashCopy where the source volumes for the FlashCopy mappings are the secondary volumes of the one Metro Mirror relationship. A backup is taken from the target volumes. We establish the clone of the Metro Mirror secondary using the background copy process, and we use IFC to refresh this clone to minimize the load on the system and to quickly establish the clone. Establishing the clone in a short time decreases the interval when no full clone of the volume is available, thus enhancing our recovery time objective.

6.3.1 Checking Volumes and the Metro Mirror relationship

To check the volumes to be used for our FlashCopy mapping, we use the command **`lsvdisk -delim :`** as shown in Example 6-25. To eliminate the white space between the output columns we use the **`-delim`** switch.

Example 6-25 lsvdisk -delim

```
IBM_2145:itsosvccl3:admin>lsvdisk -delim :
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
0:vd-back-l2-0001:0:io_grp0:online:1:DS4500:25.0GB:striped:::::60050768018E80E8E00
0000000000017:0:1
1:vd-back-l2-0002:0:io_grp0:online:1:DS4500:25.0GB:striped:::::60050768018E80E8E00
0000000000018:0:1
2:vd-prod-l2-0001:0:io_grp0:online:0:RemoteCopyMDG2:25.0GB:striped:::2:ITSOSRV01_R
EL01:60050768018E80E8E000000000000014:0:1
3:vd-prod-l2-0002:0:io_grp0:online:0:RemoteCopyMDG2:25.0GB:striped:::3:ITSOSRV01_R
EL02:60050768018E80E8E000000000000015:0:1
IBM_2145:itsosvccl3:admin>
```

We use the volumes whose names contain the characters `prod` as the source volume for the FlashCopy mapping and the volumes whose names contain the characters `back` as the target volumes.

For example, we display the properties of the volume with the ID 0, using the command **`lsvdisk -delim : 0`** as shown in Example 6-26.

Example 6-26 Details of Volume 0

```
IBM_2145:itsosvccl3:admin>lsvdisk -delim : 0
id:0
name:vd-back-l2-0001
IO_group_id:0
IO_group_name:io_grp0
status:online
mdisk_grp_id:1
mdisk_grp_name:DS4500
capacity:25.0GB
type:striped
```

```
formatted:no
mdisk_id:
mdisk_name:
FC_id:
FC_name:
RC_id:
RC_name:
vdisk_UID:60050768018E80E8E000000000000017
throttling:0
preferred_node_id:2
fast_write_state:empty
cache:readwrite
udid:
fc_map_count:0
sync_rate:50
copy_count:1

copy_id:0
status:online
sync:yes
primary:yes
mdisk_grp_id:1
mdisk_grp_name:DS4500
type:striped
mdisk_id:
mdisk_name:
fast_write_state:empty
used_capacity:25.00GB
real_capacity:25.00GB
free_capacity:0.00MB
overallocation:100
autoexpand:
warning:
grainsize:
IBM_2145:itsosvccl3:admin>
```

To display the one Metro Mirror Consistency Group, we issue the command **lsrcconsistgrp -delim :** as shown in Example 6-27.

Example 6-27 Existing Remote Copy Consistency Group

```
IBM_2145:itsosvccl3:admin>lsrcconsistgrp -delim :
id:name:master_cluster_id:master_cluster_name:aux_cluster_id:aux_cluster_name:prim
ary:state:relationship_count:copy_type
255:ITSO_SRV_GRP:0000020066403A44:itsosvccl2:0000020063A03A38:itsosvccl3:master:co
nsistent_synchronized:2:metro
IBM_2145:itsosvccl3:admin>
```

As shown, only one Consistency Group exists. To show the properties of only this Consistency Group, we specify the ID with the command (Example 6-28).

Example 6-28 Properties of the Consistency Group with ID 255

```
IBM_2145:itsosvccl3:admin>lsrcconsistgrp -delim : 255
id:255
name:ITSO_SRV_GRP
```

```
master_cluster_id:0000020066403A44
master_cluster_name:itsosvccl2
aux_cluster_id:0000020063A03A38
aux_cluster_name:itsosvccl3
primary:master
state:consistent_synchronized
relationship_count:2
freeze_time:
status:online
sync:
copy_type:metro
RC_rel_id:2
RC_rel_name:ITSOSRV01_REL01
RC_rel_id:3
RC_rel_name:ITSOSRV01_REL02
IBM_2145:itsosvccl3:admin>
```

The displayed information verifies that we have successfully set up the Remote Copy Consistency Group, which is associated with two Metro Mirror relationships.

We can also verify the properties of the two Metro Mirror relationships by issuing the command **lsrcrelationship -delim : 2**. One of the two relationships is shown in Example 6-29.

Example 6-29 Properties of the relationship with ID 0

```
IBM_2145:itsosvccl3:admin>lsrcrelationship -delim : 2
id:2
name:ITSOSRV01_REL01
master_cluster_id:0000020066403A44
master_cluster_name:itsosvccl2
master_vdisk_id:8
master_vdisk_name:vd-prod-l1-0001
aux_cluster_id:0000020063A03A38
aux_cluster_name:itsosvccl3
aux_vdisk_id:2
aux_vdisk_name:vd-prod-l2-0001
primary:master
consistency_group_id:255
consistency_group_name:ITSO_SRV_GRP
state:consistent_synchronized
bg_copy_priority:50
progress:
freeze_time:
status:online
sync:
copy_type:metro
IBM_2145:itsosvccl3:admin>
```

The other relationship is shown in Example 6-30.

Example 6-30 Properties of the relationship with ID 3

```
IBM_2145:itsosvccl3:admin>lsrcrelationship -delim : 3
id:3
name:ITSOSRV01_REL02
```

```
master_cluster_id:0000020066403A44
master_cluster_name:itsosvccl2
master_vdisk_id:9
master_vdisk_name:vd-prod-l1-0002
aux_cluster_id:0000020063A03A38
aux_cluster_name:itsosvccl3
aux_vdisk_id:3
aux_vdisk_name:vd-prod-l2-0002
primary:master
consistency_group_id:255
consistency_group_name:ITSO_SRV_GRP
state:consistent_synchronized
bg_copy_priority:50
progress:
freeze_time:
status:online
sync:
copy_type:metro
IBM_2145:itsosvccl3:admin>
```

The volumes of our Metro Mirror relationship in location 1 (production location) are named:

- `vd-prod-l1-0001`
- `vd-prod-l1-0002`

In location 2 (business continuity location), the volumes are named:

- `vd-prod-l2-0001`
- `vd-prod-l2-0002`

We use the two volumes that serve as the secondary volumes for the Metro Mirror relationship in location 2 as the source volumes for the FlashCopy mappings.

6.3.2 Creating FlashCopy mappings

We set up the FlashCopy mappings and Consistency Group in the following order:

1. Create FlashCopy mappings.
2. Create a FlashCopy Consistency Group.
3. Add FlashCopy mappings to the Consistency Group.

> **Alternate method:** Another method is to create the Consistency Group first. Then, create the FlashCopy mappings and add the mappings to the Consistency Group in the same step.

We create the FlashCopy mappings using the command **mkfcmap**. The syntax is shown in Example 6-31.

Example 6-31 The mkfcmap command syntax

```
>>- -- -- mkfcmap -- ----------------------------------->
>-- -source --+- src_vdisk_id ---+-- -------------------------->
              '- src_vdisk_name -'
>-- -target --+- target_vdisk_id ---+-- ----------------------->
              '- target_vdisk_name -'
>--+-----------------------+-- ----------------------------->
   '- -name -- new_name_arg -'
>--+---------------------------------------+-- -------------->
   '- -consistgrp --+- consist_group_id ---+-'
                    '- consist_group_name -'
>--+----------------------+-- --+--------------+------------->
   '- -copyrate -- percent -'     '- -autodelete -'
>--+-----------------------+--+---------------+-------------->
   '- -grainsize --+- 64 --+-'  '- -incremental -'
                   '- 256 -'
>--+---------------------------+----------------------------->
   '- -cleanrate ---- percent ---'
>--+---------------------------+--------------------------><
   '- -iogrp --+- iogroup_name -+-'
               '- iogroup_id  --'
```

We do not use every optional parameter; for instance, we use the default grain size at 256 KB. In our example, we create the first FlashCopy mapping (Example 6-32).

Example 6-32 Create first FlashCopy mapping

```
IBM_2145:itsosvccl3:admin>mkfcmap -source vd-prod-l2-0001 -target vd-back-l2-0001
-name fcm-back-001 -copyrate 70 -incremental
FlashCopy Mapping, id [0], successfully created
IBM_2145:itsosvccl3:admin>
```

We then create the second FlashCopy mapping (Example 6-33).

Example 6-33 Create second FlashCopy mapping

```
IBM_2145:itsosvccl3:admin>mkfcmap -source vd-prod-l2-0002 -target vd-back-l2-0002
-name fcm-back-002 -copyrate 70
FlashCopy Mapping, id [1], successfully created
IBM_2145:itsosvccl3:admin>
```

We review the creation of both FlashCopy mappings using the command **lsfcmap -delim :** as shown in Example 6-34.

Example 6-34 Review settings for both FlashCopy mappings

```
IBM_2145:itsosvccl3:admin>lsfcmap -delim :
id:name:source_vdisk_id:source_vdisk_name:target_vdisk_id:target_vdisk_name:group_
id:group_name:status:progress:copy_rate:clean_progress:incremental
0:fcm-back-001:2:vd-prod-l2-0001:0:vd-back-l2-0001:::idle_or_copied:0:70:100:on
1:fcm-back-002:3:vd-prod-l2-0002:1:vd-back-l2-0002:::idle_or_copied:0:70:100:off
IBM_2145:itsosvccl3:admin>
```

When reviewing the FlashCopy mappings, we realize that we did not include the **-incremental** parameter in our second FlashCopy mapping. We cannot modify this FlashCopy mapping to be incremental, so we have to delete the FlashCopy mapping and recreate it. (We can modify a FlashCopy mapping using the command **chfcmap**.) We delete the FlashCopy mapping using the command **rmfcmap** (Example 6-35). We only show the syntax here.

Example 6-35 The rmfcmap command syntax

```
>>- -- -- rmfcmap -- --+----------+-- ------------------>
                               '- -force -'
>--+- fc_map_id ---+------------------------------------------><
   '- fc_map_name -'
```

After we recreate the second FlashCopy mapping, this time using the flag **-incremental**, we view the properties of that FlashCopy mapping using the command **lsfcmap 1** (Example 6-36).

Example 6-36 Properties of the FlashCopy mapping with ID 1

```
IBM_2145:itsosvccl3:admin>lsfcmap 1
id 1
name fcm-back-002
source_vdisk_id 3
source_vdisk_name vd-prod-l2-0002
target_vdisk_id 1
target_vdisk_name vd-back-l2-0002
group_id
group_name
status idle_or_copied
progress 0
copy_rate 70
start_time
dependent_mappings 0
autodelete off
clean_progress 100
clean_rate 50
incremental on
difference 100
grain_size 256
IO_group_id 0
IO_group_name io_grp0
IBM_2145:itsosvccl3:admin>
```

Using **lsvdisk** again, as shown in Example 6-37, we see that all four volumes are part of FlashCopy mappings.

Example 6-37 All four volumes are part of FlashCopy mappings

```
IBM_2145:itsosvccl3:admin>lsvdisk -delim :
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
0:vd-back-l2-0001:0:io_grp0:online:1:DS4500:25.0GB:striped:0:fcm-back-001:::600507
68018E80E8E000000000000017:1:1
1:vd-back-l2-0002:0:io_grp0:online:1:DS4500:25.0GB:striped:1:fcm-back-002:::600507
68018E80E8E000000000000018:1:1
2:vd-prod-l2-0001:0:io_grp0:online:0:RemoteCopyMDG2:25.0GB:striped:0:fcm-back-001:
2:ITSOSRV01_REL01:60050768018E80E8E000000000000014:1:1
3:vd-prod-l2-0002:0:io_grp0:online:0:RemoteCopyMDG2:25.0GB:striped:1:fcm-back-002:
3:ITSOSRV01_REL02:60050768018E80E8E000000000000015:1:1
IBM_2145:itsosvccl3:admin>
```

6.3.3 Creating a Consistency Group

At this point, we create a FlashCopy Consistency Group in which to put the FlashCopy mappings. We use the **mkfcconsistgrp** command, and the syntax is shown in Example 6-38.

Example 6-38 The mkfcconsistgrp command syntax

```
>>- -- -- mkfcconsistgrp -- ---------------------------->
>--+------------------------------+--+---------------+--------><
   '- -name -- consist_group_name -'  '- -autodelete -'
```

Example 6-39 shows the actual command that we used to create our Consistency Group.

Example 6-39 Consistency Group successfully created

```
IBM_2145:itsosvccl3:admin>mkfcconsistgrp -name fcg-back-001
FlashCopy Consistency Group, id [3], successfully created
IBM_2145:itsosvccl3:admin>
```

Example 6-40 shows the syntax of the command **lsfcconsistgrp**, which displays the created Consistency Groups.

Example 6-40 The lsfcconsistgrp command syntax

```
>>- -- -- lsfcconsistgrp -- ---------------------------->
>--+-----------------------------------+-- --+----------+-- ---->
   '- -filtervalue -- attribute=value -'     '- -nohdr -'
>--+----------------------+-- -- --+-----------------+--------->
   '- -delim -- delimiter -'        '- -filtervalue? -'
>--+---------------+------------------------------------------><
   +- object_id ---+
   '- object_name -'
```

We use the **lsfcconsistgrp** command to verify the creation of the Consistency Group that we created (Example 6-41).

Example 6-41 Empty Consistency Group

```
IBM_2145:itsosvccl3:admin>lsfcconsistgrp
id              name            status
3               fcg-back-001    empty
IBM_2145:itsosvccl3:admin>
```

At this point, we have two FlashCopy mappings and one empty FlashCopy Consistency Group.

6.3.4 Adding FlashCopy mappings to a Consistency Group

Before we can start the FlashCopy, we include the two FlashCopy mappings in the Consistency Group that we just created. We include them by issuing the command **chfcmap**. The syntax is shown in Example 6-42.

Example 6-42 The chfcmap command syntax

```
>>- -- -- chfcmap -- --+-------------------------+-- --->
                               '- -name -- new_name_arg -'
>--+---------+-- ---------------------------------------------->
   '- -force -'
>--+---------------------------------------+-- --------------->
   '- -consistgrp --+- consist_group_id ---+-'
                    '- consist_group_name -'
>--+----------------------+-- --+------------------------+--->
   '- -copyrate  -- percent-'     '- -autodelete  --+-on--+-'
                                                     '-off-'
>--+----------------------------+--+- fc_map_id ---+---------><
   '- -cleanrate ---- percent ---'  '- fc_map_name -'
```

We put the FlashCopy mappings into the Consistency Group using the command shown in Example 6-43.

Example 6-43 Associating both FlashCopy mappings with the same Consistency Group

```
IBM_2145:itsosvccl3:admin>chfcmap -consistgrp fcg-back-001 fcm-back-001
IBM_2145:itsosvccl3:admin>chfcmap -consistgrp fcg-back-001 fcm-back-002
IBM_2145:itsosvccl3:admin>
```

To view the Consistency Group, we issue the command **lsfcconsistgrp** again (Example 6-44).

Example 6-44 Consistency Group in Idle_or_Copied state

```
IBM_2145:itsosvccl3:admin>lsfcconsistgrp
id              name            status
3               fcg-back-001    idle_or_copied
IBM_2145:itsosvccl3:admin>
```

The Consistency Group state changed from `Empty` to `Idle_or_Copied`. We get the properties, including the FlashCopy mappings that are now associated with the Consistency Group, when we specify the ID of the group with the command (Example 6-45).

Example 6-45 Properties of Consistency Group with ID 3

```
IBM_2145:itsosvccl3:admin>lsfcconsistgrp 3
id 3
name fcg-back-001
status idle_or_copied
autodelete off
FC_mapping_id 0
FC_mapping_name fcm-back-001
FC_mapping_id 1
FC_mapping_name fcm-back-002
IBM_2145:itsosvccl3:admin>
```

Example 6-46 shows the two FlashCopy mappings by using the command **`lsfcmap`**. We see here that both FlashCopy mappings are members of the same Consistency Group.

Example 6-46 Both FlashCopy mappings associated with same Consistency Group

```
IBM_2145:itsosvccl3:admin>lsfcmap -delim :
id:name:source_vdisk_id:source_vdisk_name:target_vdisk_id:target_vdisk_name:group_
id:group_name:status:progress:copy_rate:clean_progress:incremental
0:fcm-back-001:2:vd-prod-l2-0001:0:vd-back-l2-0001:3:fcg-back-001:idle_or_copied:0
:70:100:on
1:fcm-back-002:3:vd-prod-l2-0002:1:vd-back-l2-0002:3:fcg-back-001:idle_or_copied:0
:70:100:on
IBM_2145:itsosvccl3:admin>
```

6.3.5 Preparing a FlashCopy mapping

We want to prepare the FlashCopy Consistency Group (and thus the FlashCopy mappings associated with it) prior to issuing the start of the group. We issue the command **`prestartfcconsistgrp`**, and the syntax is shown in Example 6-47.

Example 6-47 The prestartfcconsistgrp command syntax

```
>>- -- -- prestartfcconsistgrp -- ---------------------->
>--+- fc_consist_group_id ---+---------------------------------><
   '- fc_consist_group_name -'
```

We prepare the group, and then we immediately issue the command to verify that the state of the Consistency Group is Prepared (Example 6-48).

Example 6-48 Prepare and view group

```
IBM_2145:itsosvccl3:admin>prestartfcconsistgrp fcg-back-001
IBM_2145:itsosvccl3:admin>lsfcconsistgrp
id              name             status
3               fcg-back-001     prepared
IBM_2145:itsosvccl3:admin>
```

We see the state of the FlashCopy mappings by using the command **lsfcmap** (Example 6-49).

Example 6-49 Both FlashCopy mappings in Prepared state

```
IBM_2145:itsosvccl3:admin>lsfcmap -delim :
id:name:source_vdisk_id:source_vdisk_name:target_vdisk_id:target_vdisk_name:group_
id:group_name:status:progress:copy_rate:clean_progress:incremental
0:fcm-back-001:2:vd-prod-l2-0001:0:vd-back-l2-0001:3:fcg-back-001:prepared:0:70:10
0:on
1:fcm-back-002:3:vd-prod-l2-0002:1:vd-back-l2-0002:3:fcg-back-001:prepared:0:70:10
0:on
IBM_2145:itsosvccl3:admin>
```

6.3.6 Starting a Consistency Group

Example 6-50 shows the syntax of the command **startfcconsistgrp**.

Example 6-50 The startfcconsistgrp command syntax

```
>>- -- -- startfcconsistgrp -- --+---------+-- --------->
                                   '- -prep -'
>--+- fc_consist_group_id ---+---------------------------------><
   '- fc_consist_group_name -'
```

The -prep option: You can prepare the Consistency Group first by using the **-prep** option, but because in our case we have already prepared the group, we do not need to do it now.

Example 6-51 shows how we start the Consistency Group (using the ID to specify it). Immediately after starting the Consistency Group, we view its status.

Example 6-51 Start and view Consistency Group

```
IBM_2145:itsosvccl3:admin>startfcconsistgrp 3
IBM_2145:itsosvccl3:admin>lsfcconsistgrp
id              name            status
3               fcg-back-001    copying
IBM_2145:itsosvccl3:admin>
```

At this point, the state of the Consistency Group is shown as `Copying`, as expected. In Example 6-52, we see the state of both FlashCopy mappings is also shown as `Copying`.

Example 6-52 Both FlashCopy mappings in Copying state

```
IBM_2145:itsosvccl3:admin>lsfcmap -delim :
id:name:source_vdisk_id:source_vdisk_name:target_vdisk_id:target_vdisk_name:group_
id:group_name:status:progress:copy_rate:clean_progress:incremental
0:fcm-back-001:2:vd-prod-l2-0001:0:vd-back-l2-0001:3:fcg-back-001:copying:1:70:100
:on
1:fcm-back-002:3:vd-prod-l2-0002:1:vd-back-l2-0002:3:fcg-back-001:copying:1:70:100
:on
IBM_2145:itsosvccl3:admin>
```

Example 6-53 shows the details of the FlashCopy mapping with the ID 0.

Example 6-53 The lsfcmap -delim : 0 command

```
IBM_2145:itsosvccl3:admin>lsfcmap -delim : 0
id:0
name:fcm-back-001
source_vdisk_id:2
source_vdisk_name:vd-prod-l2-0001
target_vdisk_id:0
target_vdisk_name:vd-back-l2-0001
group_id:3
group_name:fcg-back-001
status:copying
progress:3
copy_rate:70
start_time:080807161152
dependent_mappings:0
autodelete:off
clean_progress:100
clean_rate:50
incremental:on
difference:96
grain_size:256
IO_group_id:0
IO_group_name:io_grp0
IBM_2145:itsosvccl3:admin>
```

We see that the state of the FlashCopy mapping is `copying` and that the progress is only 3% because we just started it, using a background copy rate of 70%.

Example 6-54 shows the details of one of the volumes.

Example 6-54 The lsvdisk -delim : 2 command

```
IBM_2145:itsosvccl3:admin>lsvdisk -delim : 2
id:2
name:vd-prod-l2-0001
IO_group_id:0
IO_group_name:io_grp0
status:online
mdisk_grp_id:0
mdisk_grp_name:RemoteCopyMDG2
capacity:25.0GB
type:striped
formatted:no
mdisk_id:
mdisk_name:
FC_id:0
FC_name:fcm-back-001
RC_id:2
RC_name:ITSOSRV01_REL01
vdisk_UID:60050768018E80E8E000000000000014
throttling:0
preferred_node_id:2
fast_write_state:empty
```

```
cache:readwrite
udid:0
fc_map_count:1
sync_rate:50
copy_count:1

copy_id:0
status:online
sync:yes
primary:yes
mdisk_grp_id:0
mdisk_grp_name:RemoteCopyMDG2
type:striped
mdisk_id:
mdisk_name:
fast_write_state:empty
used_capacity:25.00GB
real_capacity:25.00GB
free_capacity:0.00MB
overallocation:100
autoexpand:
warning:
grainsize:
IBM_2145:itsosvccl3:admin>
```

Volume **vd-prod-12-0001** is a member of both a Remote Copy Relationship and a FlashCopy mapping.

After the background copy process finishes copying the data to the target volume, the state changes to `Idle_or_Copied` (Example 6-55).

Example 6-55 Idle_or_Copied state

```
IBM_2145:ITSOCL3:admin>lsfcconsistgrp 1
id 1
name fcg-back-001
status idle_or_copied
FC_mapping_id 0
FC_mapping_name fcm-back-001
FC_mapping_id 1
FC_mapping_name fcm-back-002
IBM_2145:ITSOCL3:admin>
```

We can also review the details of one of the FlashCopy mappings (Example 6-56).

Example 6-56 The lsfcmap -delim : 0 command

```
IBM_2145:itsosvccl3:admin>lsfcmap -delim : 0
id:0
name:fcm-back-001
source_vdisk_id:2
source_vdisk_name:vd-prod-l2-0001
target_vdisk_id:0
target_vdisk_name:vd-back-l2-0001
group_id:3
group_name:fcg-back-001
status:idle_or_copied
progress:100
copy_rate:70
start_time:080807161152
dependent_mappings:0
autodelete:off
clean_progress:100
clean_rate:50
incremental:on
difference:0
grain_size:256
IO_group_id:0
IO_group_name:io_grp0
IBM_2145:itsosvccl3:admin>
```

Because we set up the FlashCopy mappings as incremental, we can prepare and start the Consistency Group again to refresh it, and only the grains that have changed since the last FlashCopy process are copied. This approach puts less load on the cluster and makes the full clone of the volumes available faster.

6.3.7 Modifying a FlashCopy mapping

To modify a FlashCopy mapping, we issue the command **chfcmap**, which uses the syntax that is shown in Example 6-57.

Example 6-57 The chfcmap command syntax

```
>>- -- -- chfcmap -- --+-------------------------+-- --->
                               '- -name -- new_name_arg -'
>--+----------+-- -------------------------------------------->
   '- -force -'
>--+----------------------------------------+-- --------------->
   '- -consistgrp --+- consist_group_id ---+-'
                    '- consist_group_name -'
>--+----------------------+-- --+-------------------------+--->
   '- -copyrate  -- percent-'     '- -autodelete  --+-on--+-'
                                                    '-off-'
>--+---------------------------+--+- fc_map_id ---+----------><
   '- -cleanrate ---- percent ---'  '- fc_map_name -'
```

We want to change both FlashCopy mappings to a higher background copy priority. Example 6-58 on page 223 shows how we change the background copy priority to 90 and verify the settings for the FlashCopy mapping with `ID 0`. We make the same changes with the second FlashCopy mapping.

Example 6-58 Change the copy priority to 90

```
IBM_2145:itsosvccl3:admin>chfcmap -copyrate 90
IBM_2145:itsosvccl3:admin>lsfcmap 0
id 0
name fcm-back-001
source_vdisk_id 2
source_vdisk_name vd-prod-l2-0001
target_vdisk_id 0
target_vdisk_name vd-back-l2-0001
group_id 3
group_name fcg-back-001
status idle_or_copied
progress 100
copy_rate 90
start_time 080807161152
dependent_mappings 0
autodelete off
clean_progress 100
clean_rate 50
incremental on
difference 0
grain_size 256
IO_group_id 0
IO_group_name io_grp0
IBM_2145:itsosvccl3:admin>
```

6.3.8 Stopping a FlashCopy mapping and a Consistency Group

Our example FlashCopy Consistency Group was set up to be fully copied using background copy and to be incrementally refreshed. Its status was Idle_or_Copied. To show how to stop and remove FlashCopy mappings and Consistency Groups, we removed these FlashCopy mappings, created new mappings, and put them into the existing Consistency Group. We set up the mappings with a copy rate of 0 and then started the Consistency Group, resulting in the status shown in Example 6-59.

Example 6-59 Viewing the new FlashCopy mappings with status Copying

```
IBM_2145:itsosvccl3:admin>lsfcmap -delim :
id:name:source_vdisk_id:source_vdisk_name:target_vdisk_id:target_vdisk_name:group_
id:group_name:status:progress:copy_rate:clean_progress:incremental
0:fcm-back-001:2:vd-prod-l2-0001:0:vd-back-l2-0001:3:fcg-back-001:copying:0:0:100:
off
1:fcm-back-002:3:vd-prod-l2-0002:1:vd-back-l2-0002:3:fcg-back-001:copying:0:0:100:
off
IBM_2145:itsosvccl3:admin>
```

Before we are able to stop a FlashCopy mapping, which is associated with a Consistency Group, we need to remove it from the Consistency Group using the **`-force`** parameter.

Immediately afterwards we view the result (Example 6-60). As we see, only the FlashCopy mapping `fcm-back-002` is still part of the Consistency Group.

Example 6-60   Removing a FlashCopy mapping from a Consistency Group

```
IBM_2145:itsosvccl3:admin>chfcmap -force 0
IBM_2145:itsosvccl3:admin>lsfcmap -delim :
id:name:source_vdisk_id:source_vdisk_name:target_vdisk_id:target_vdisk_name:group_
id:group_name:status:progress:copy_rate:clean_progress:incremental
0:fcm-back-001:2:vd-prod-l2-0001:0:vd-back-l2-0001:::copying:0:0:100:off
1:fcm-back-002:3:vd-prod-l2-0002:1:vd-back-l2-0002:3:fcg-back-001:copying:0:0:100:
off
IBM_2145:itsosvccl3:admin>
```

Example 6-61 shows the syntax of the command **stopfcmap**, which is used to stop a FlashCopy mapping.

Example 6-61   The stopfcmap command syntax

```
>>- -- -- stopfcmap -- --+---------+-- ----------------->
                                 '- -force-'

>--+- fc_map_id ---+------------------------------------------><
   '- fc_map_name -'

```

We now stop the FlashCopy mapping `fcm-back-001` and view the result immediately (Example 6-62).

Example 6-62   Stopping a stand-alone FlashCopy mapping

```
IBM_2145:itsosvccl3:admin>stopfcmap 0
IBM_2145:itsosvccl3:admin>lsfcmap -delim :
id:name:source_vdisk_id:source_vdisk_name:target_vdisk_id:target_vdisk_name:group_
id:group_name:status:progress:copy_rate:clean_progress:incremental
0:fcm-back-001:2:vd-prod-l2-0001:0:vd-back-l2-0001:::stopped:0:0:100:off
1:fcm-back-002:3:vd-prod-l2-0002:1:vd-back-l2-0002:3:fcg-back-001:copying:0:0:100:
off
IBM_2145:itsosvccl3:admin>
```

To stop a Consistency Group, we use the command **stopfcconsistgrp**, for which the syntax is shown in Example 6-63.

Example 6-63   The stopfcconsistgrp command syntax

```
>>- -- -- stopfcconsistgrp -- --+---------+-- ---------->
                                        '- -force-'
>--+- fc_consist_group_id  --+---------------------------------><
   '- fc_consist_group_name -'
```

We now stop the Consistency Group and view the result (Example 6-64).

Example 6-64   Stopping a Consistency Group

```
IBM_2145:itsosvccl3:admin>stopfcconsistgrp 3
IBM_2145:itsosvccl3:admin>lsfcconsistgrp
id               name              status
```

```
3               fcg-back-001      stopped
IBM_2145:itsosvccl3:admin>
```

6.3.9 Deleting a FlashCopy mapping and a Consistency Group

The target volumes of the FlashCopy mappings in the Stopped state are `offline`, as shown in Example 6-65.

Example 6-65 Viewing the volumes of the FlashCopy mappings: target volumes are offline

```
IBM_2145:itsosvccl3:admin>lsvdisk -delim :
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
0:vd-back-l2-0001:0:io_grp0:offline:1:DS4500:25.0GB:striped:0:fcm-back-001:::60050
768018E80E8E000000000000017:1:1
1:vd-back-l2-0002:0:io_grp0:offline:1:DS4500:25.0GB:striped:1:fcm-back-002:::60050
768018E80E8E000000000000018:1:1
2:vd-prod-l2-0001:0:io_grp0:online:0:RemoteCopyMDG2:25.0GB:striped:0:fcm-back-001:
2:ITSOSRV01_REL01:60050768018E80E8E000000000000014:1:1
3:vd-prod-l2-0002:0:io_grp0:online:0:RemoteCopyMDG2:25.0GB:striped:1:fcm-back-002:
3:ITSOSRV01_REL02:60050768018E80E8E000000000000015:1:1
IBM_2145:itsosvccl3:admin>
```

Example 6-66 shows the syntax of the command **rmfcmap**, which is used to delete a FlashCopy mapping.

Example 6-66 The rmfcmap command syntax

```
>>- -- -- rmfcmap -- --+----------+-- ------------------>
                        '- -force -'
>--+- fc_map_id ---+------------------------------------------><
   '- fc_map_name -'
```

The FlashCopy mapping can only be deleted after it is in the stopped or Idle_or_Copied state. Thus, in case the FlashCopy mapping is in any other state, it must first finish its background copy process to complete the copy and enter the state Idle_or_Copied, or it must be stopped and enter the Stopped state.

If it is in the Stopped state, we must use the -**force** parameter, which (with the **rmfcmap** command) forces the target volume to go online. We have to force the target volume to go online, because stopping the FlashCopy mapping out of any state without having finished the background copy process invalidates the content of the target volume that is brought online with that invalid content. We now delete the FlashCopy mapping `fcm-back-001` and show the result (there is one less FlashCopy mapping in the list), as shown in Example 6-67.

Example 6-67 Delete the FlashCopy mapping

```
IBM_2145:itsosvccl3:admin>rmfcmap -force 0
IBM_2145:itsosvccl3:admin>lsfcmap -delim :
id:name:source_vdisk_id:source_vdisk_name:target_vdisk_id:target_vdisk_name:group_
id:group_name:status:progress:copy_rate:clean_progress:incremental
1:fcm-back-002:3:vd-prod-l2-0002:1:vd-back-l2-0002:3:fcg-back-001:stopped:0:0:100:
off
IBM_2145:itsosvccl3:admin>
```

Example 6-68 shows the syntax of the command **rmfcconsistgrp**, which is used to delete a FlashCopy Consistency Group.

Example 6-68 The rmfcconsistgrp command syntax

```
>>- -- -- rmfcconsistgrp -- --+----------+-- ----------->
                                '- -force -'
>--+- fc_consist_group_id ---+---------------------------------><
   '- fc_consist_group_name -'
```

Because the Consistency Group is not empty, we need to force the delete (Example 6-69).

Example 6-69 Delete a Consistency Group: forced delete

```
IBM_2145:itsosvccl3:admin>rmfcconsistgrp -force 3
IBM_2145:itsosvccl3:admin>lsfcconsistgrp
IBM_2145:itsosvccl3:admin>
```

6.4 FlashCopy - GUI operation

In this section we describe the tasks that we can perform at a FlashCopy level by using the GUI. If we have a small number of mappings it is often easier to control it using the GUI. When using many mappings, you might find it easier to use the CLI (see more details in 6.2, “FlashCopy - CLI operation” on page 192) to execute your commands.

Important: It is possible for many users to be logged into the GUI at any given time. However, no locking mechanism exists, so be aware that if two users change the same object at the same time, the last action entered from the GUI is the one that will take effect. Data entries made through the GUI are case sensitive.

6.4.1 Introduction to the GUI

The Cluster Home panel (Figure 6-2 on page 227) is an important panel and is referred to as the Home panel throughout this chapter.

Figure 6-2 Home panel

From this Home panel, on the left panel, there is a dynamic menu.

Dynamic menu

This new version of the Cluster GUI includes a new dynamic menu that is located in the left column of the window. To navigate using this menu, move the mouse over the various icons and choose a page that you want to display (Figure 6-3 on page 228).

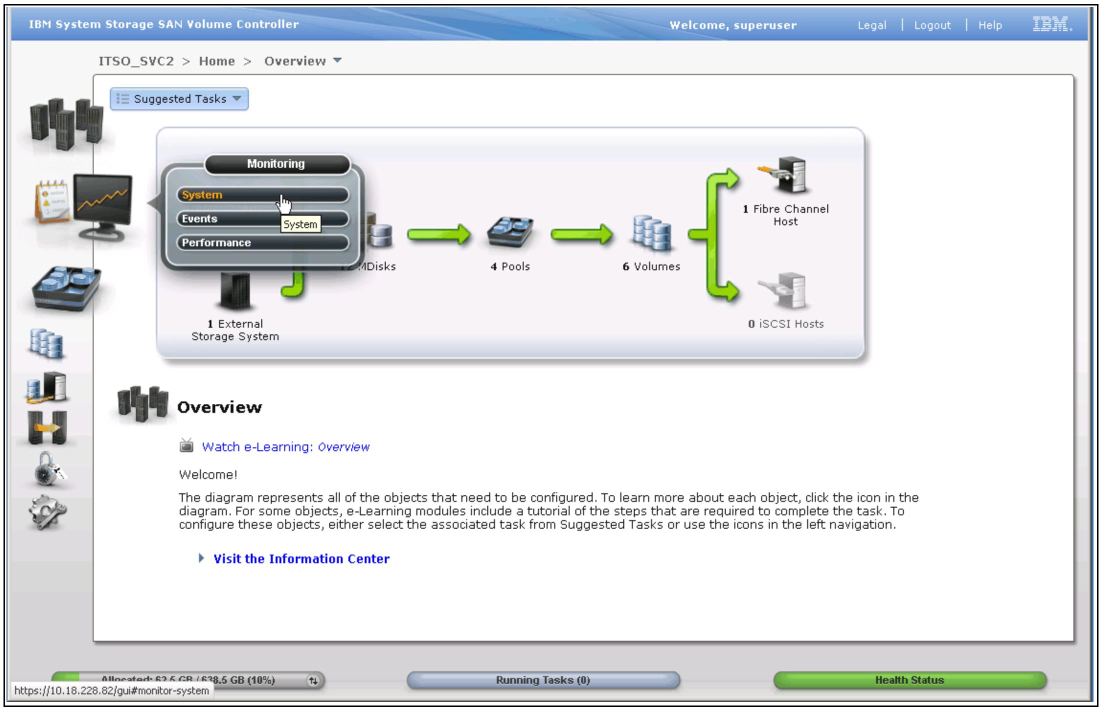


Figure 6-3 The dynamic menu in the left column

A non-dynamic version of this menu exists for slow connections. To access the non-dynamic menu, select **Low graphics mode**, as shown in Figure 6-4 and Figure 6-5 on page 229.

Figure 6-4 The SVC GUI Login panel

Figure 6-4 displays the Storwize V7000 Low graphics mode.

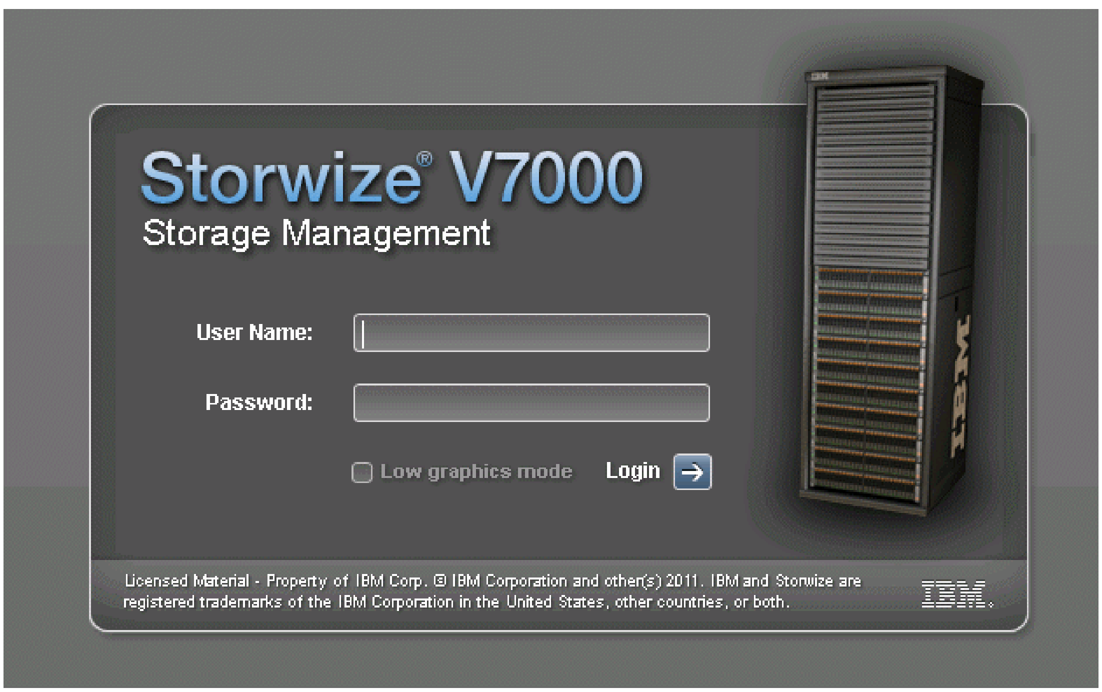


Figure 6-5 The V7000 GUI Login panel

Figure 6-6 shows the non-dynamic version of the menu.

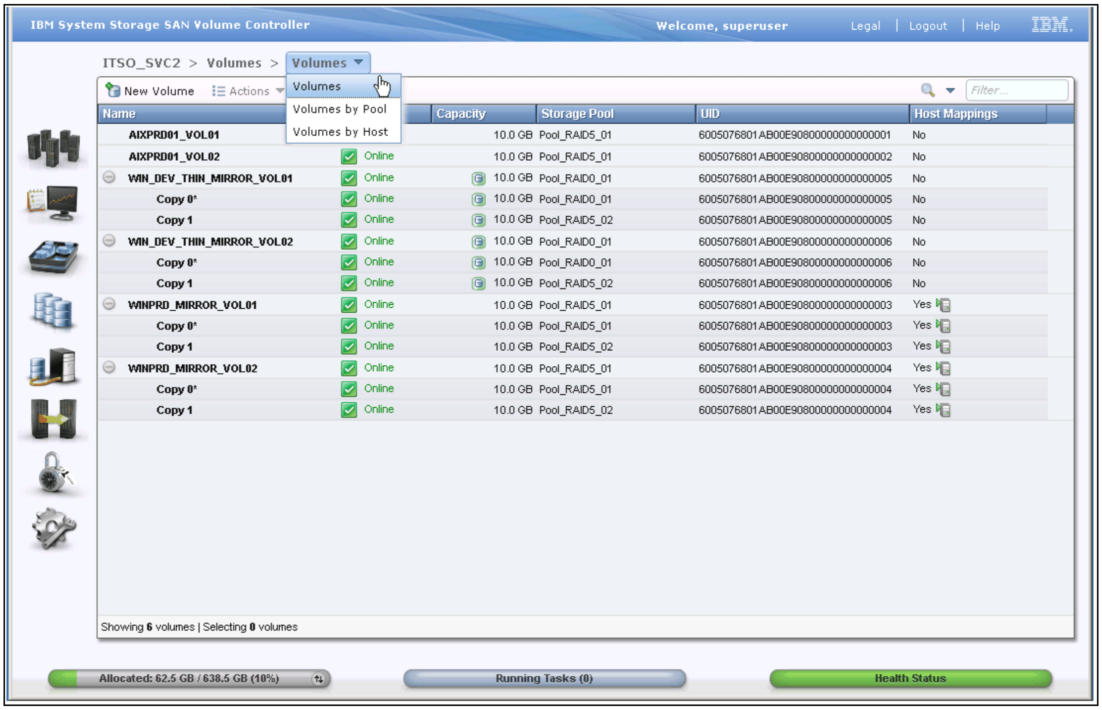


Figure 6-6 Non-dynamic menu in the left column

In this case, in the upper part of the page, there is a pull-down menu for navigating between submenus. For example, in Figure 6-6, Volumes, Volumes by Pool, and Volumes by Host are submenus (pull-down menus) for the Volumes menu.

There are three ways to visualize and manage your FlashCopy:

- By using the FlashCopy panel (Figure 6-7)

  The IBM FlashCopy function copies the contents of a source to a target volume. Any data that existed on the target volume is lost and is replaced by the copied data.

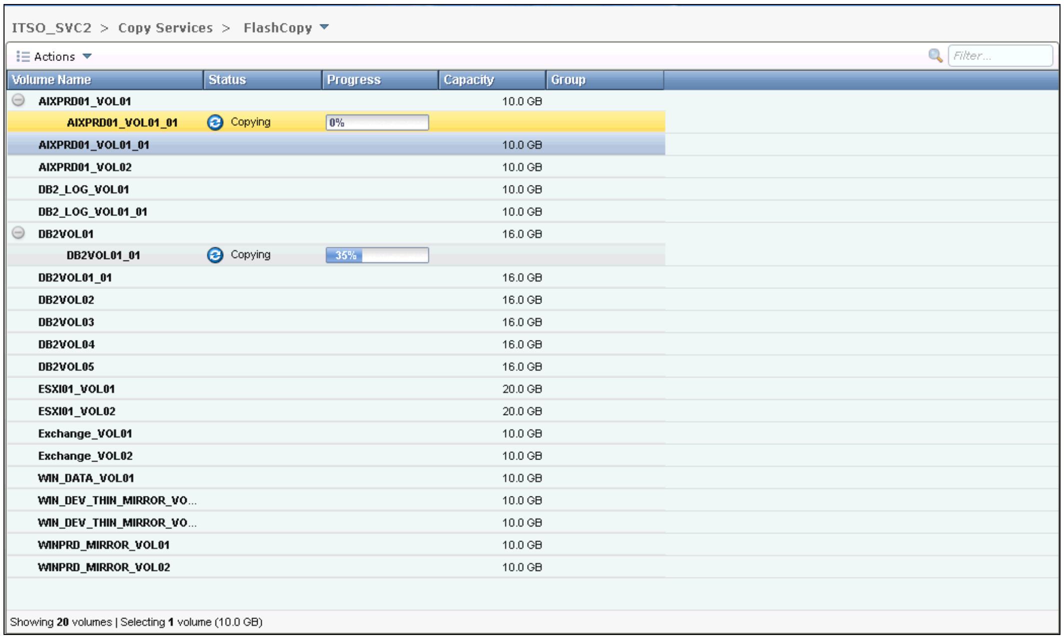


Figure 6-7 FlashCopy panel

- By using the Consistency Groups panel (Figure 6-8)

  A Consistency Group is a container for mappings. You can add many mappings to a Consistency Group.

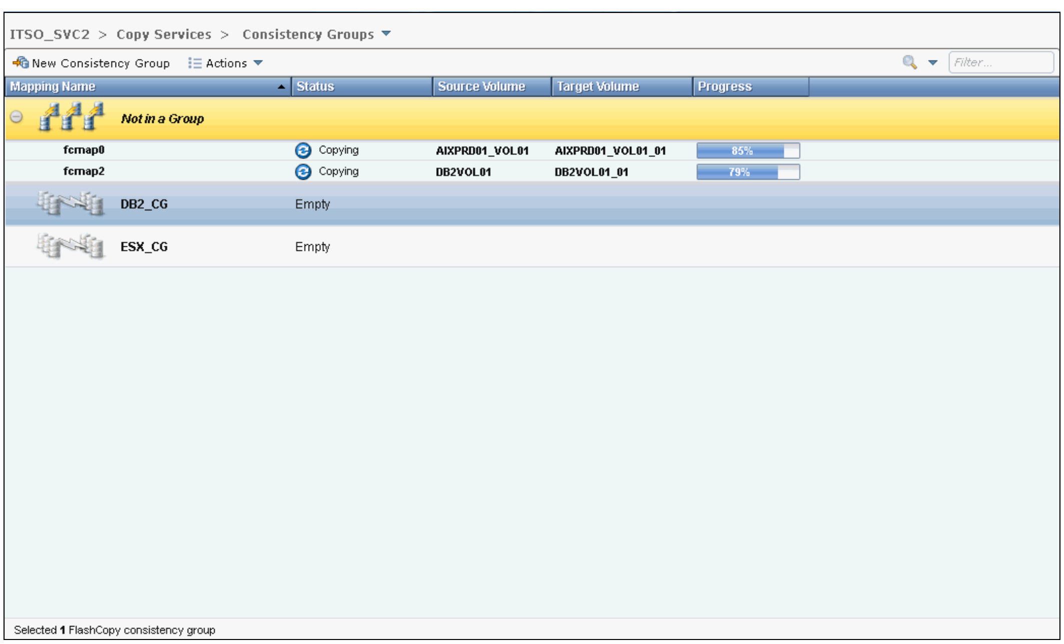


Figure 6-8 Consistency Groups panel

- By using the FlashCopy Mappings panel (Figure 6-9)

  A FlashCopy mapping defines the relationship between a source volume and a target volume.

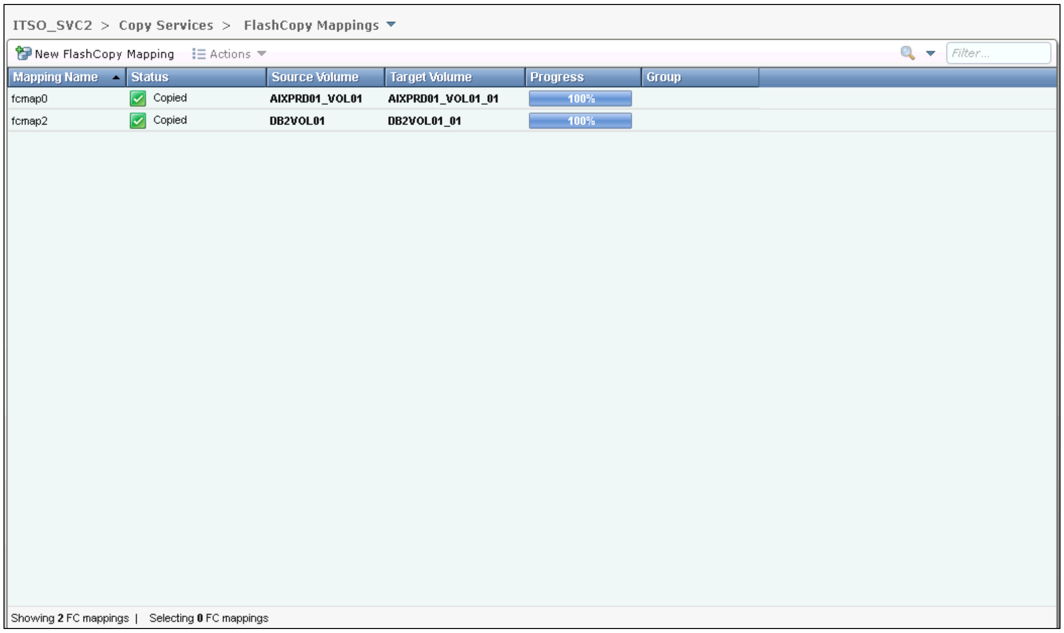


Figure 6-9 FlashCopy Mappings panel

6.4.2 Creating a FlashCopy mapping

In this section, we create FlashCopy mappings for volumes with their respective targets.

To perform this action, follow these steps:

1. From the Cluster Welcome panel, click **Copy Services** → **FlashCopy**. The FlashCopy panel opens (Figure 6-10 on page 233).

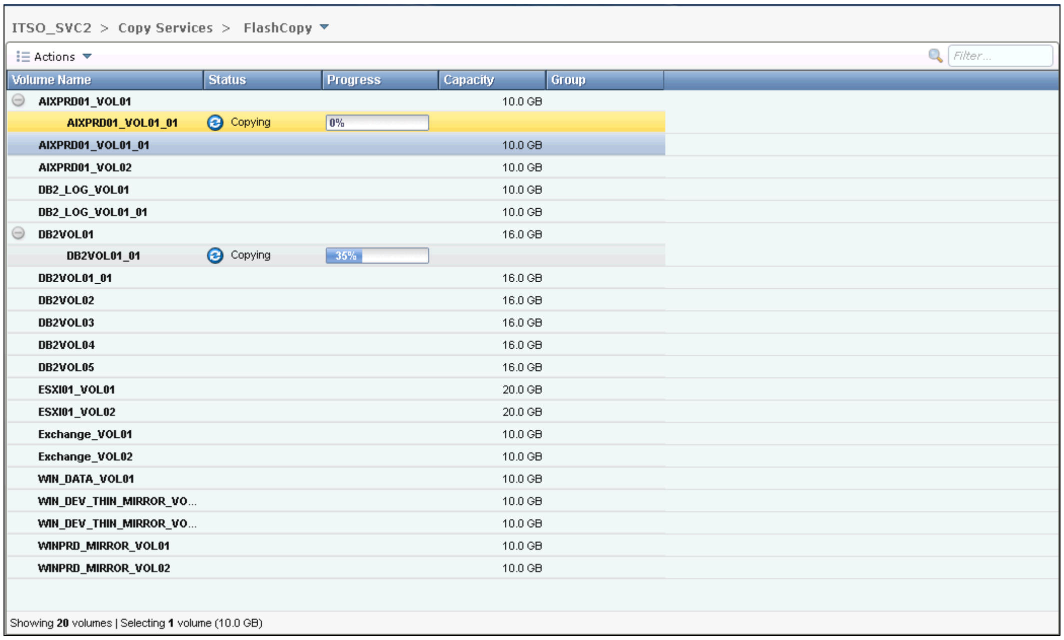


Figure 6-10 FlashCopy panel

2. We select the volume where we want to create the FlashCopy relationship (Figure 6-11 on page 234).

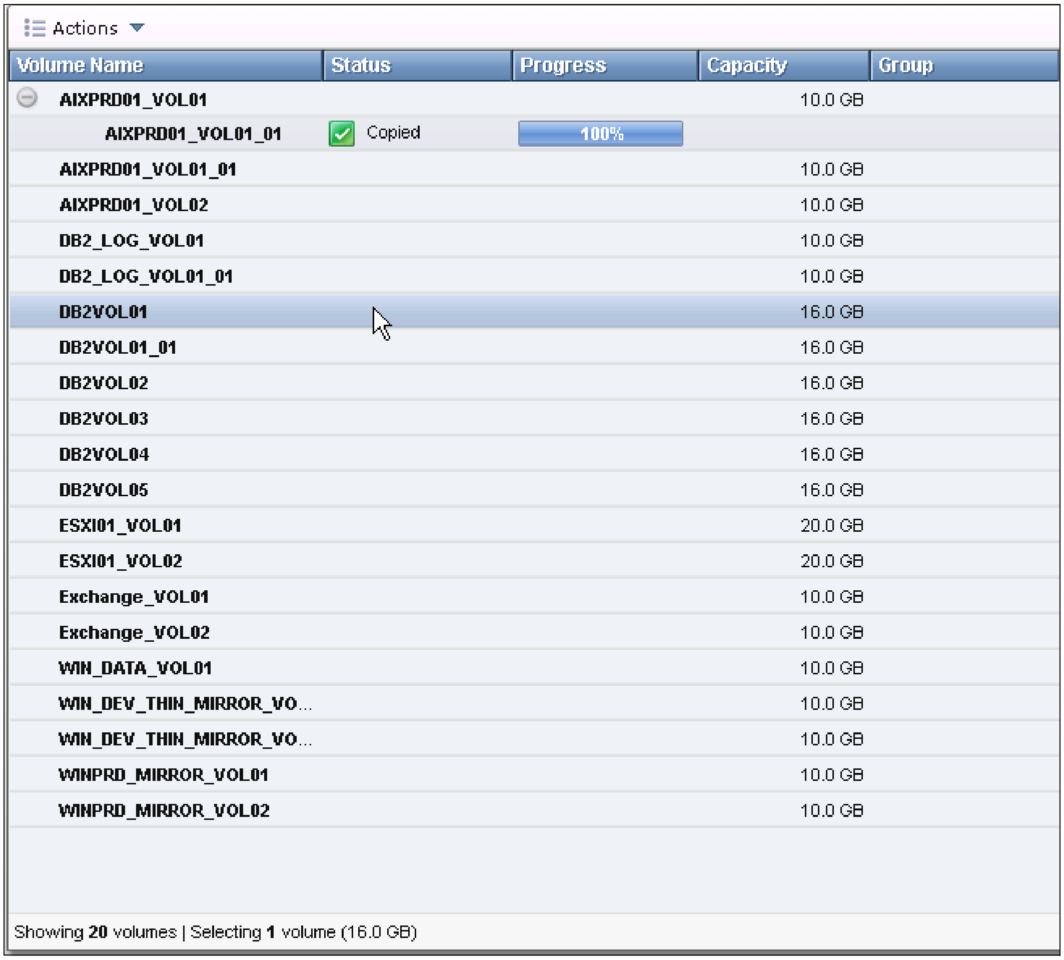


Figure 6-11 FlashCopy mapping: Select the volume

Multiple FlashCopy mappings: To create multiple FlashCopy mappings at one time, select multiple volumes by holding down the **Ctrl** key and using the mouse to select the entries that you want. Figure 6-12 on page 235 illustrates the multiple volume selection.

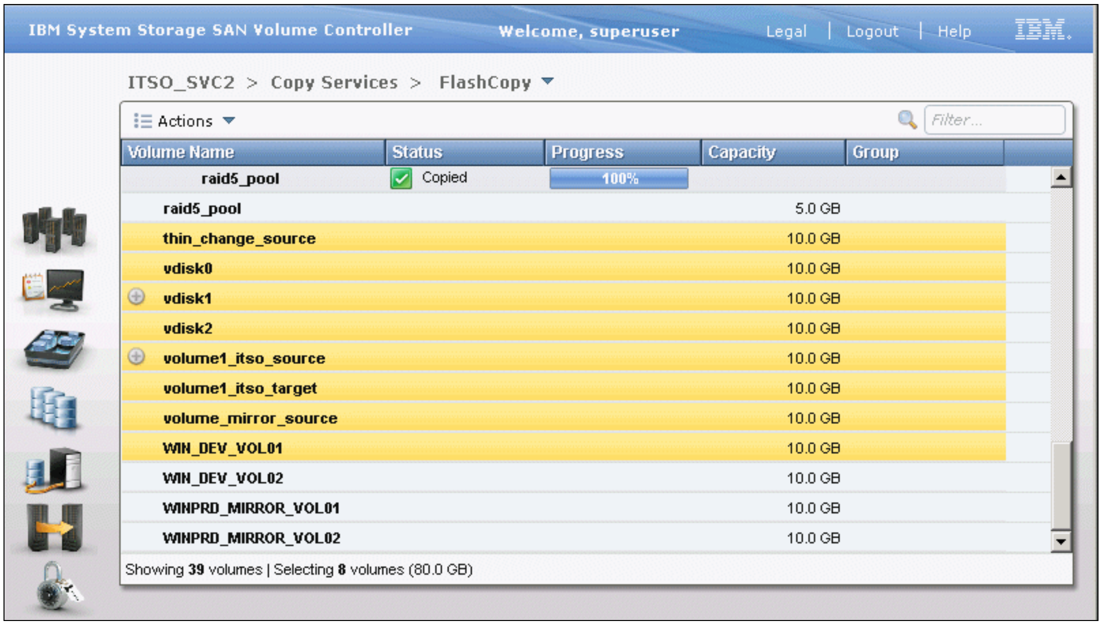


Figure 6-12 Multiple FlashCopy mapping selection

At this point, depending on whether you have already created the target volumes for your FlashCopy mappings, there are two options:

- If you have already created the target volumes, see "Using existing target volumes" on page 236, for the steps to follow.
- If you want cluster to create the target volumes for you, see "Creating new target volumes" on page 240, for the steps to follow.

Using existing target volumes

Follow these steps to use existing target volumes for the FlashCopy mappings:

1. Select the target volume to use and click **Actions** → **Advanced FlashCopy** → **Use existing target volumes** (Figure 6-13 on page 236).

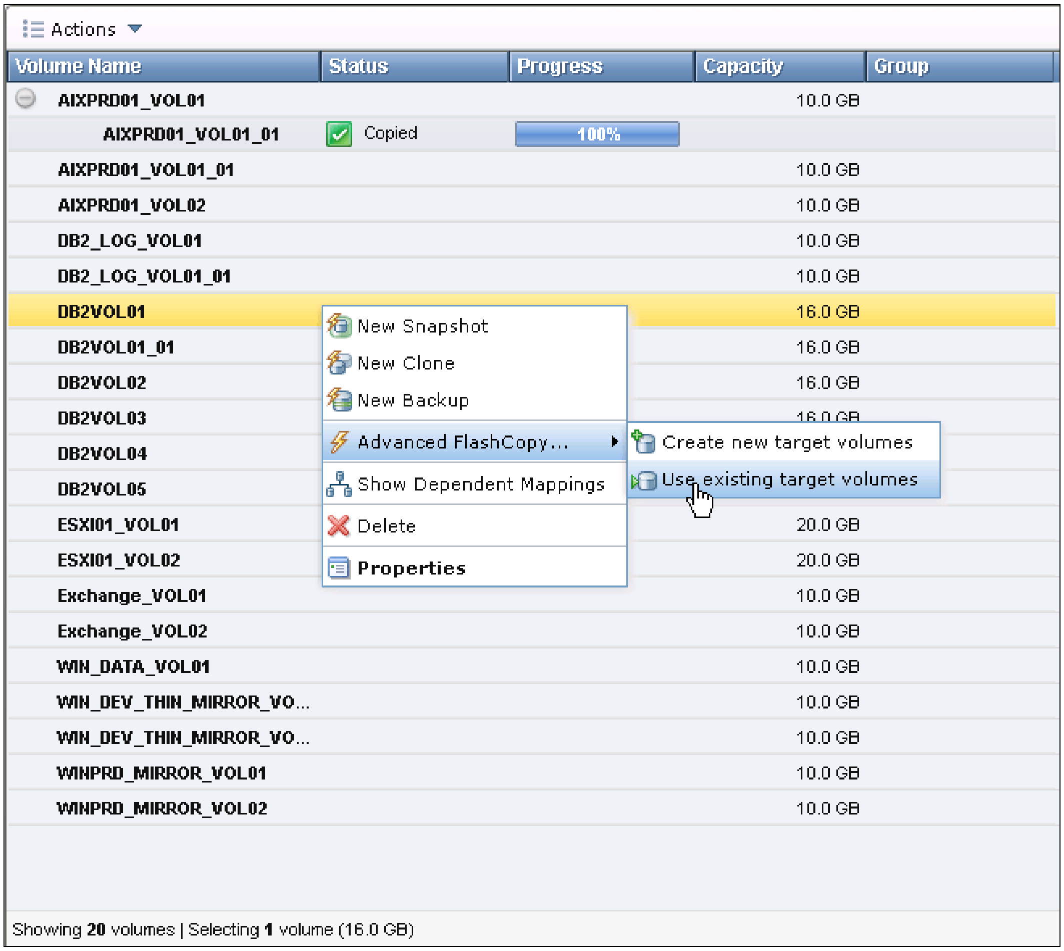


Figure 6-13 Use existing target volumes

2. The New FlashCopy Mapping window opens (see Figure 6-14 on page 237). In this window, you have to create the relationship between the source volume (the disk that is copied) and the target volume (the disk that receives the copy). A mapping can be created between any two volumes in a cluster. Select a source volume and a target volume for your FlashCopy mapping, and then click **Add**. If you need to create other copies, repeat this action.

> **Important:** The source and target volumes must be of equal size. So for a given source volume, only targets of the same size are visible in the list box.

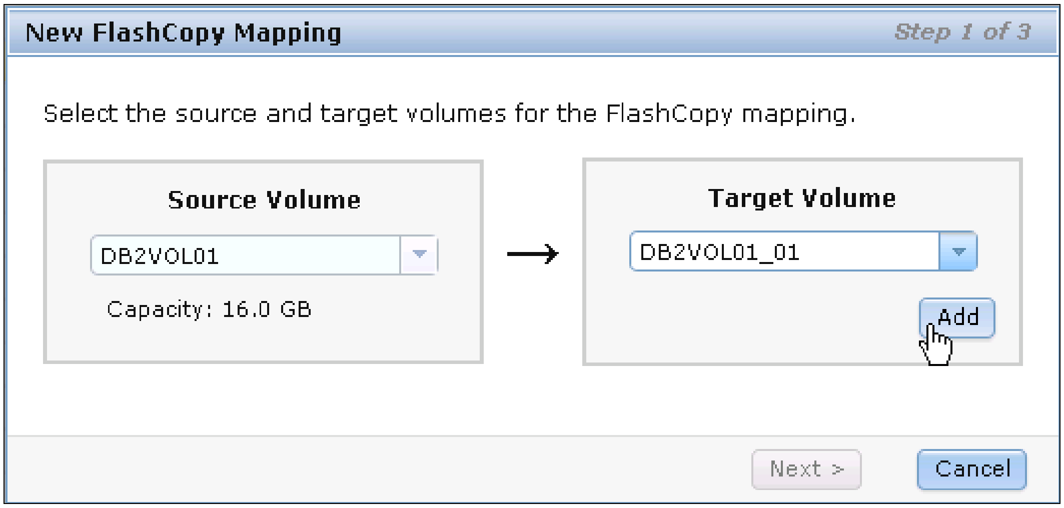


Figure 6-14 New FlashCopy Mapping

To remove a relationship that you have created, use  icon (Figure 6-15).

> **Volumes consideration:** To create a FlashCopy mapping, the volumes do not have to be in the same I/O Group or storage pool.

3. Click **Next** after you have created all of the relationships that you want to create (Figure 6-15).

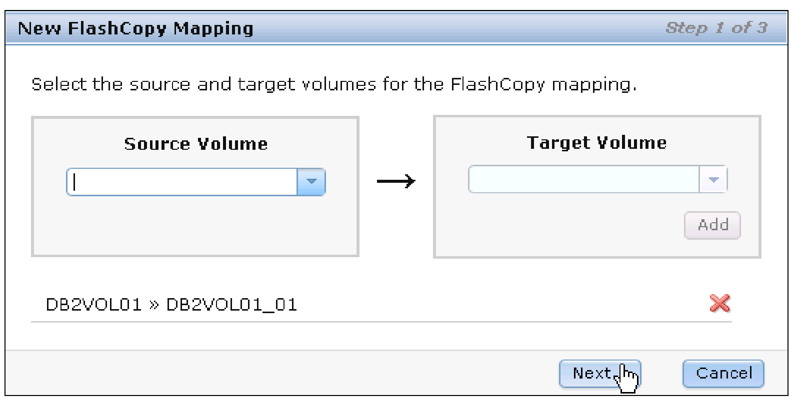


Figure 6-15 New FlashCopy Mapping

4. On the next window, select *one* FlashCopy preset. The GUI interface provides three presets (Snapshot, Clone, or Backup) to simplify the more common FlashCopy operations (Figure 6-16 on page 238).

   The presets and their use cases are described here:

   Snapshot This preset creates a copy-on-write point-in-time copy.

   Clone This preset creates an exact replica of the source volume on a target volume. The copy can be changed without affecting the original volume.

   Backup This preset creates a FlashCopy mapping that can be used to recover data or objects if the system experiences data loss. These backups can be copied multiple times from source and target volumes.

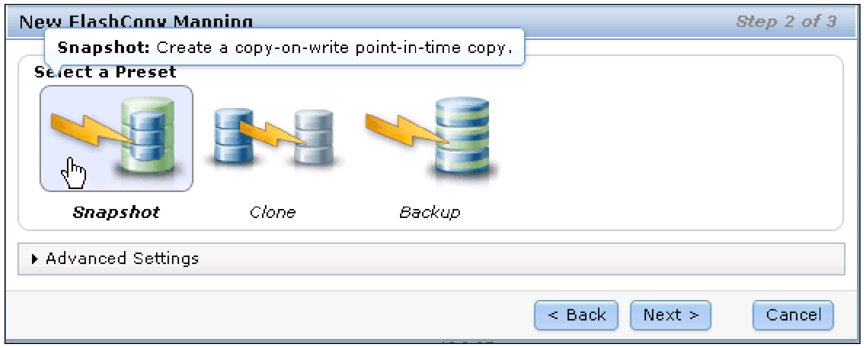


Figure 6-16 New FlashCopy Mapping window

You can customize various advanced options for whichever preset you choose. You access these settings by clicking **Advanced Settings** (Figure 6-17 on page 239).

If you prefer not to customize these settings, go directly to step 5 on page 239.

You can customize the following options, as shown in Figure 6-17 on page 239:

– Background Copy Rate: This option determines the priority that is given to the copy process. A faster rate increases the priority of the process, which can affect the performance of other operations.

– Incremental: This option copies only the parts of the source or target volumes that have changed since the last copy. Incremental copies reduce the completion time of the copy operation.

> **Incremental FlashCopy mapping:** Even if the type of the FlashCopy mapping is incremental, the first copy process copies all of the data from the source volume to the target volume.

– Delete mapping after completion: This option automatically deletes a FlashCopy mapping after the background copy is completed. Do *not* use this option when the background copy rate is set to zero (0).

– Cleaning Rate: This option minimizes the amount of time that a mapping is in the Stopping state. If the mapping has not completed, the target volume is offline while the mapping is stopping.

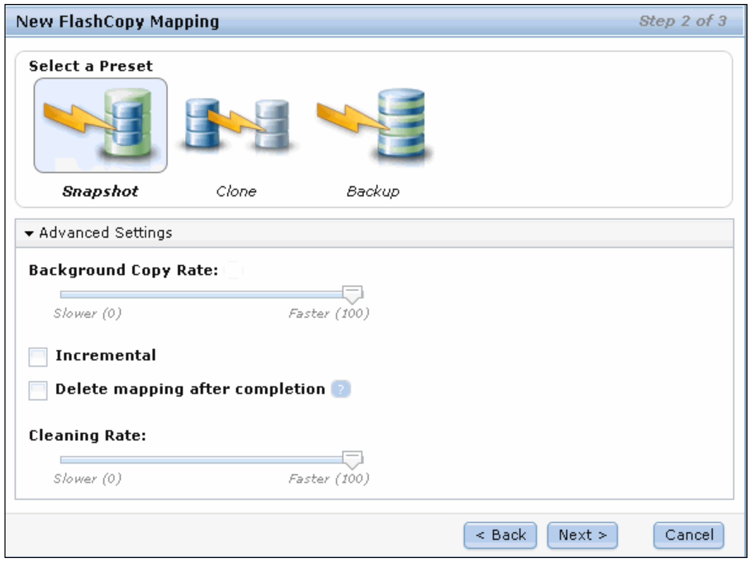


Figure 6-17   New FlashCopy Mapping Advanced Settings

5. If you want to include this FlashCopy mapping in a Consistency Group, in the next window (Figure 6-18) select **Yes, add the mappings to a consistency group**. Also select the Consistency Group from the drop-down list box.

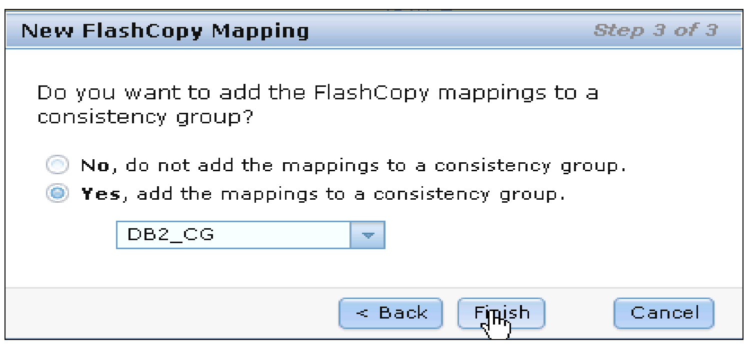


Figure 6-18   Add the mappings to a Consistency Group

If you do not want to include this FlashCopy mapping in a Consistency Group, in that window select **No, do not add the mappings to a consistency group** (Figure 6-19).

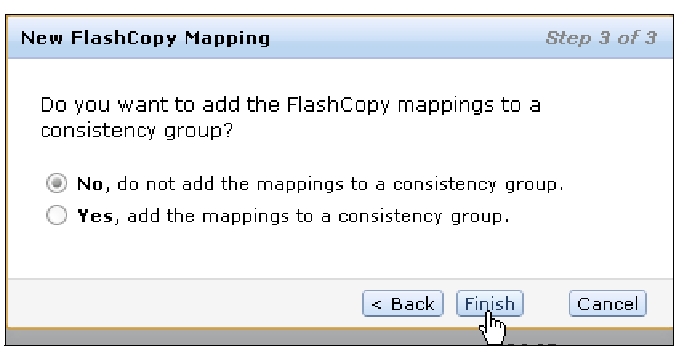


Figure 6-19   Do not add the mappings to a Consistency Group

Then, click **Finish** as shown in Figure 6-18 and Figure 6-19.

6. Check the result of this FlashCopy mapping (Figure 6-20). For each FlashCopy mapping relationship that has been created, a mapping name is automatically generated starting with `fcmap`*X*, where *X* is an available number.

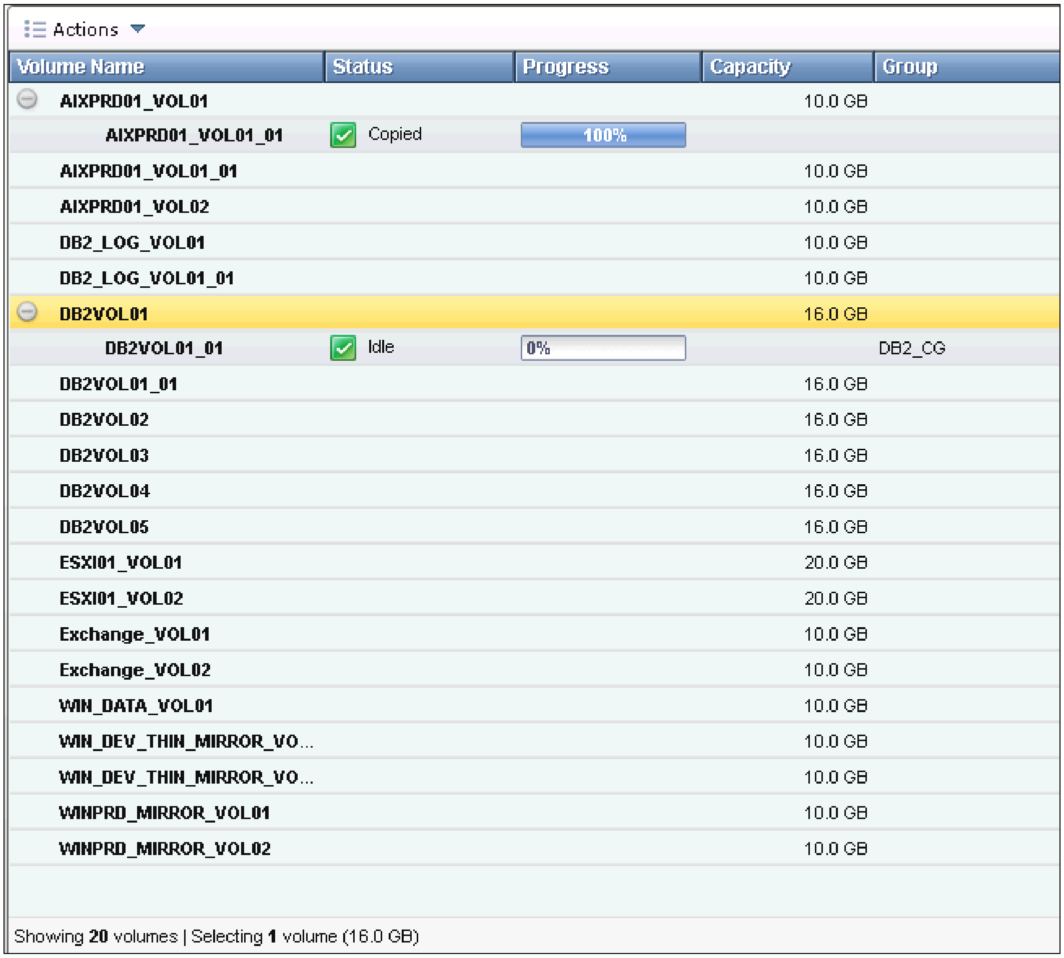


Figure 6-20 Flash Copy Mapping

At this point, the FlashCopy mapping is now ready for use.

Creating new target volumes

Follow these steps to create new target volumes for FlashCopy mapping:

1. If you do not created a target volume for this source volume, click **Actions** → **Advanced FlashCopy** → **Create new target volumes** (Figure 6-21 on page 241).

> **Target volume naming:** If the target volume does not exist, it will be created with a name based on its source volume and a generated number at the end. An example is *`source_volume_name_XX`*, where *`XX`* is a number that was generated dynamically.

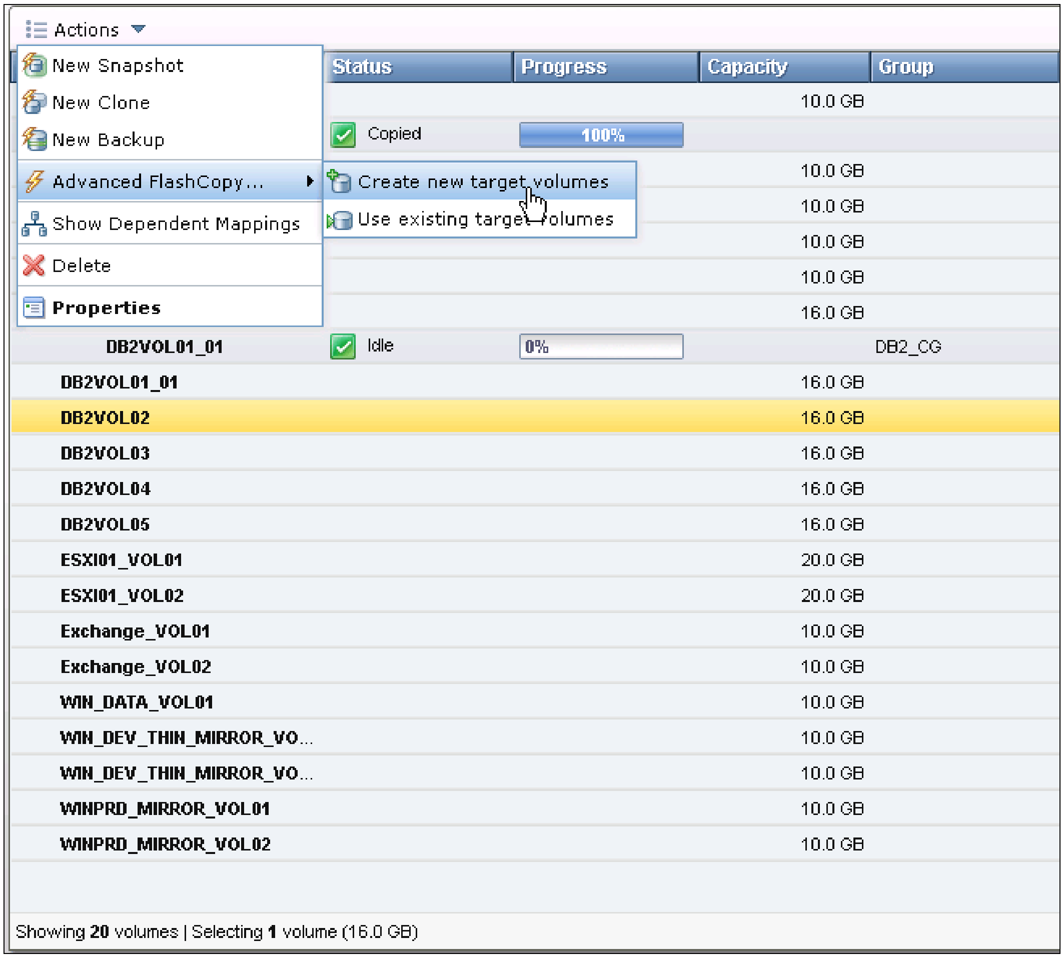


Figure 6-21 Create new target volumes action

2. On the New FlashCopy Mapping window (Figure 6-22 on page 242), you need to select *one* FlashCopy preset. The GUI interface provides three presets (Snapshot, Clone, or Backup) to simplify the more common FlashCopy operations.

   The presets and their use cases are described here:

   Snapshot This preset creates a copy-on-write point-in-time copy.

   Clone This preset creates an exact replica of the source volume on a target volume. The copy can be changed without affecting the original volume.

   Backup This preset creates a FlashCopy mapping that can be used to recover data or objects if the system experiences data loss. These backups can be copied multiple times from source and target volumes.

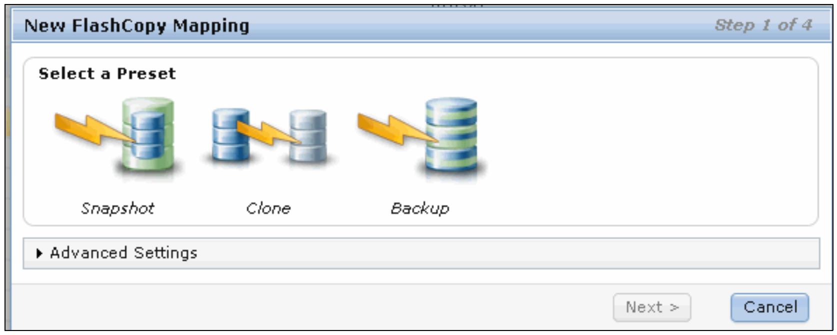


Figure 6-22 New FlashCopy Mapping window

You can customize various advanced options for whichever preset you choose. To access these settings, click **Advanced Settings** (Figure 6-23 on page 243).

If you prefer not to customize these settings, go directly to step 3 on page 243.

You can customize the following options, as shown in Figure 6-23 on page 243:

– Background Copy Rate: This option determines the priority that is given to the copy process. A faster rate increases the priority of the process, which can affect the performance of other operations.

– Incremental: This option copies only the parts of the source or target volumes that have changed since the last copy. Incremental copies reduce the completion time of the copy operation.

> **Incremental FlashCopy mapping:** Even if the type of the FlashCopy mapping is incremental, the first copy process copies all of the data from the source to the target volume.

– Delete mapping after completion: This option automatically deletes a FlashCopy mapping after the background copy is completed. Do *not* use this option when the background copy rate is set to 0.

– Cleaning Rate: This option minimizes the amount of time that a mapping is in the Stopping state. If the mapping has not completed, the target volume is offline while the mapping is stopping.

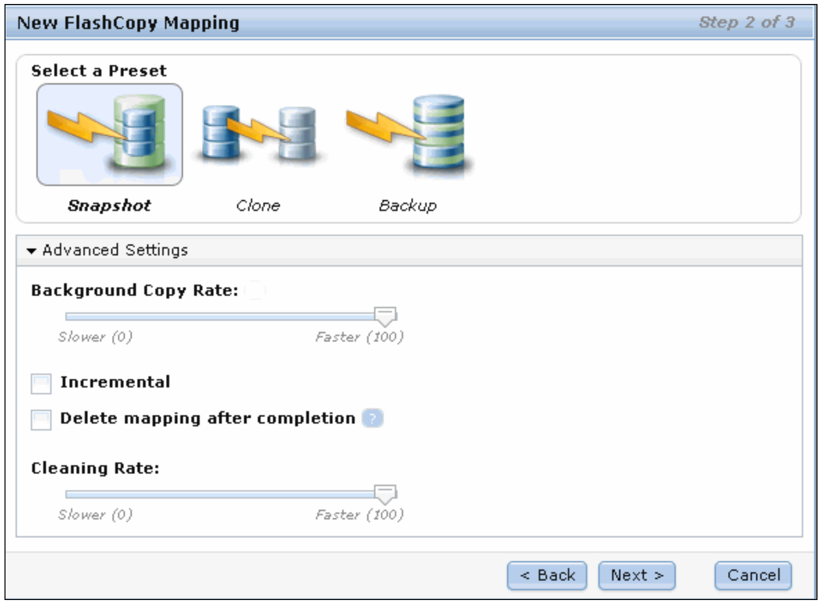


Figure 6-23 New FlashCopy Mapping Advanced Settings

3. If you want to include this FlashCopy mapping in a Consistency Group, in the next window (Figure 6-24) select **Yes, add the mappings to a consistency group**. Select the Consistency Group in the drop-down list box.

   If you do not want to include this FlashCopy mapping in a Consistency Group, select **No, do not add the mappings to a consistency group**.

   Choose whichever option you prefer, and click **Finish** (Figure 6-24).

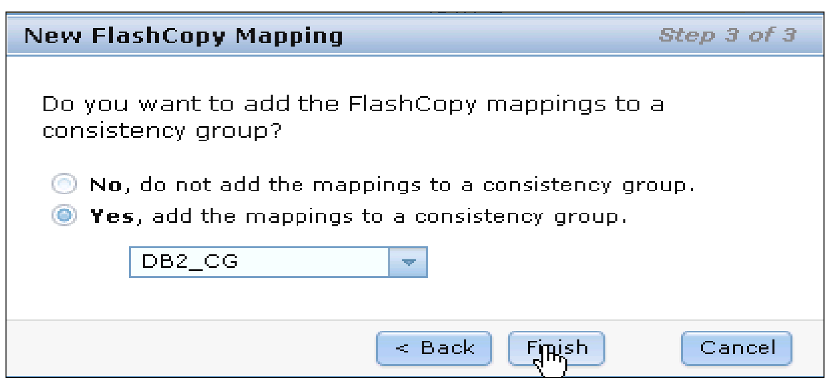


Figure 6-24 Add the mappings to a Consistency Group

4. In the next window (Figure 6-25 on page 244), select the storage pool that is used to automatically create new targets. You can choose to use the same storage pool that is

used by the source volume, or you can select a storage pool from a list. In that case, select one storage pool and then click **Next**.

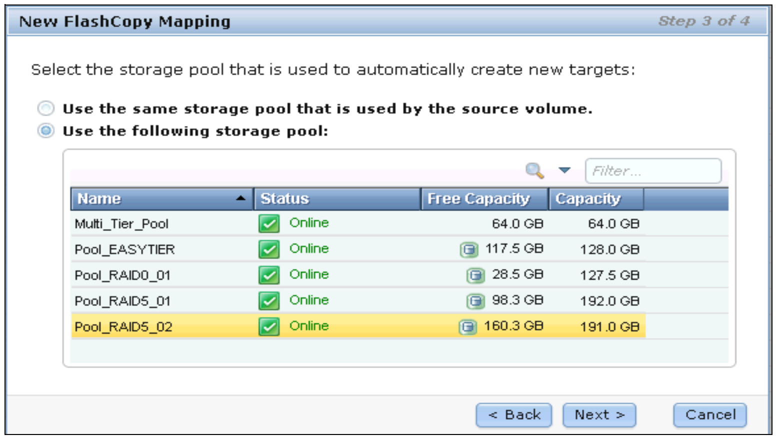


Figure 6-25 Select the storage pool

5. Select whether you want a targeted volume using thin provisioning. Three choices are available, as shown in Figure 6-26:
   - Yes, in which case, enter the following parameters:
     - Real: Type the real size that you want to allocate. This size is the amount of disk space that will actually be allocated. It can either be a percentage of the virtual size or a specific number in GBs.
     - Automatically Expand: Select auto expand, which allows the real disk size to grow as required.
     - Warning Threshold: Type a percentage or select a specific size for the usage threshold warning. This function will generate a warning when the used disk capacity on the thin-provisioned copy first exceeds the specified threshold.
   - No
   - Inherit properties from source volume, will use the same properties from source volume

Click **Finish** to complete the FlashCopy Mapping operation.

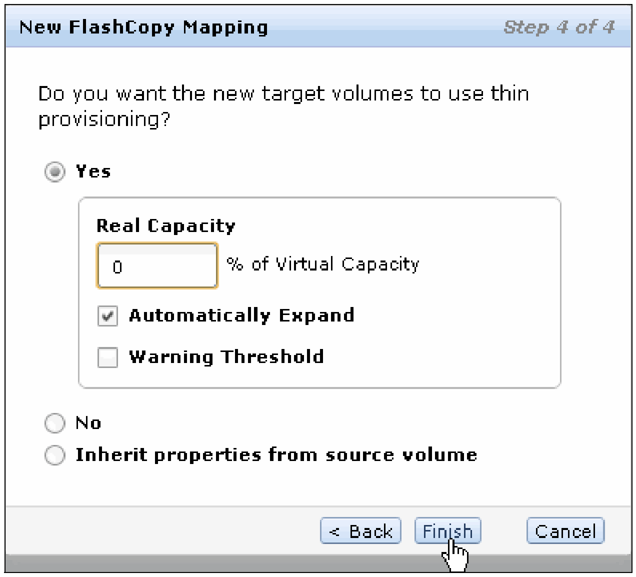
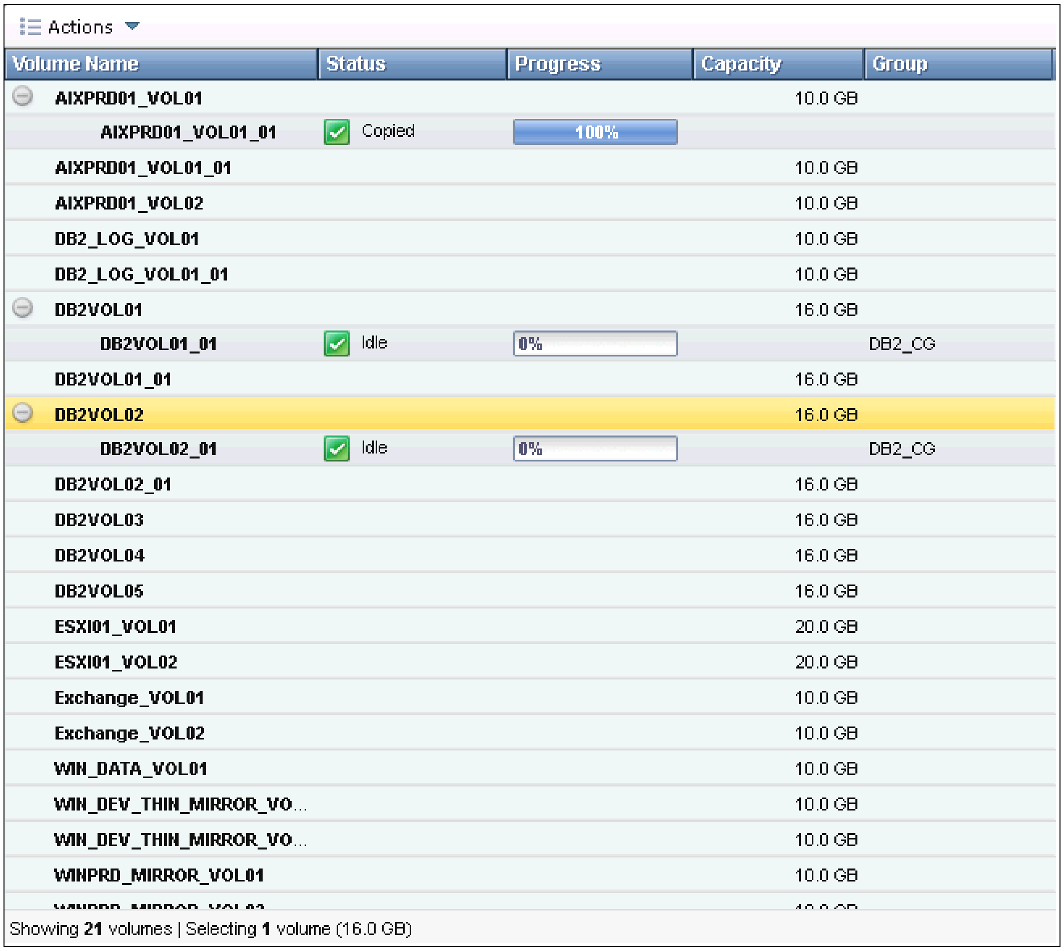


Figure 6-26 Thin-provisioning option

6. Check the result of this FlashCopy mapping (Figure 6-27). For each FlashCopy mapping relationship created, a mapping name is automatically generated starting with `fcmapX` where *X* is an available number.


Actions

Volume Name	Status	Progress	Capacity	Group
AIXPRD01_VOL01			10.0 GB	
AIXPRD01_VOL01_01	Copied	100%		
AIXPRD01_VOL01_01			10.0 GB	
AIXPRD01_VOL02			10.0 GB	
DB2_LOG_VOL01			10.0 GB	
DB2_LOG_VOL01_01			10.0 GB	
DB2VOL01			16.0 GB	
DB2VOL01_01	Idle	0%		DB2_CG
DB2VOL01_01			16.0 GB	
DB2VOL02			16.0 GB	
DB2VOL02_01	Idle	0%		DB2_CG
DB2VOL02_01			16.0 GB	
DB2VOL03			16.0 GB	
DB2VOL04			16.0 GB	
DB2VOL05			16.0 GB	
ESXI01_VOL01			20.0 GB	
ESXI01_VOL02			20.0 GB	
Exchange_VOL01			10.0 GB	
Exchange_VOL02			10.0 GB	
WIN_DATA_VOL01			10.0 GB	
WIN_DEV_THIN_MIRROR_VO...			10.0 GB	
WIN_DEV_THIN_MIRROR_VO...			10.0 GB	
WINPRD_MIRROR_VOL01			10.0 GB	

Showing **21** volumes | Selecting **1** volume (16.0 GB)

Figure 6-27 FlashCopy mapping

At this point, the FlashCopy mapping is ready for use.

Tip: You can invoke FlashCopy from the SVC/V7000 GUI, but using the SVC/V7000 GUI might be impractical if you plan to handle a large number of FlashCopy mappings or Consistency Groups periodically, or at varying times. In these cases, creating a script by using the CLI might be more convenient.

6.4.3 Creating and starting a snapshot preset with a single click

Snapshot preset: The snapshot creates a point-in-time view of production data. The snapshot is not intended to be an independent copy. Instead, it is used to maintain a view of the production data at the time that the snapshot is created. Therefore, the snapshot holds only the data from regions of the production volume that have changed since the snapshot was created. Because the snapshot preset uses thin provisioning, only the capacity that is required for the changes is used.

Snapshot uses these preset parameters:

- No background copy.
- Incremental: No
- Delete after completion: No
- Cleaning rate: No
- The target pool is the primary copy source pool.

To create and start a snapshot with one click, perform the following steps:

1. From the SVC/V7000 Welcome panel, click **Copy Services** in the left menu and then click **FlashCopy**.
2. Select the volume that you want to snapshot.

3. Click **Actions** → **New Snapshot** (Figure 6-28 on page 248).

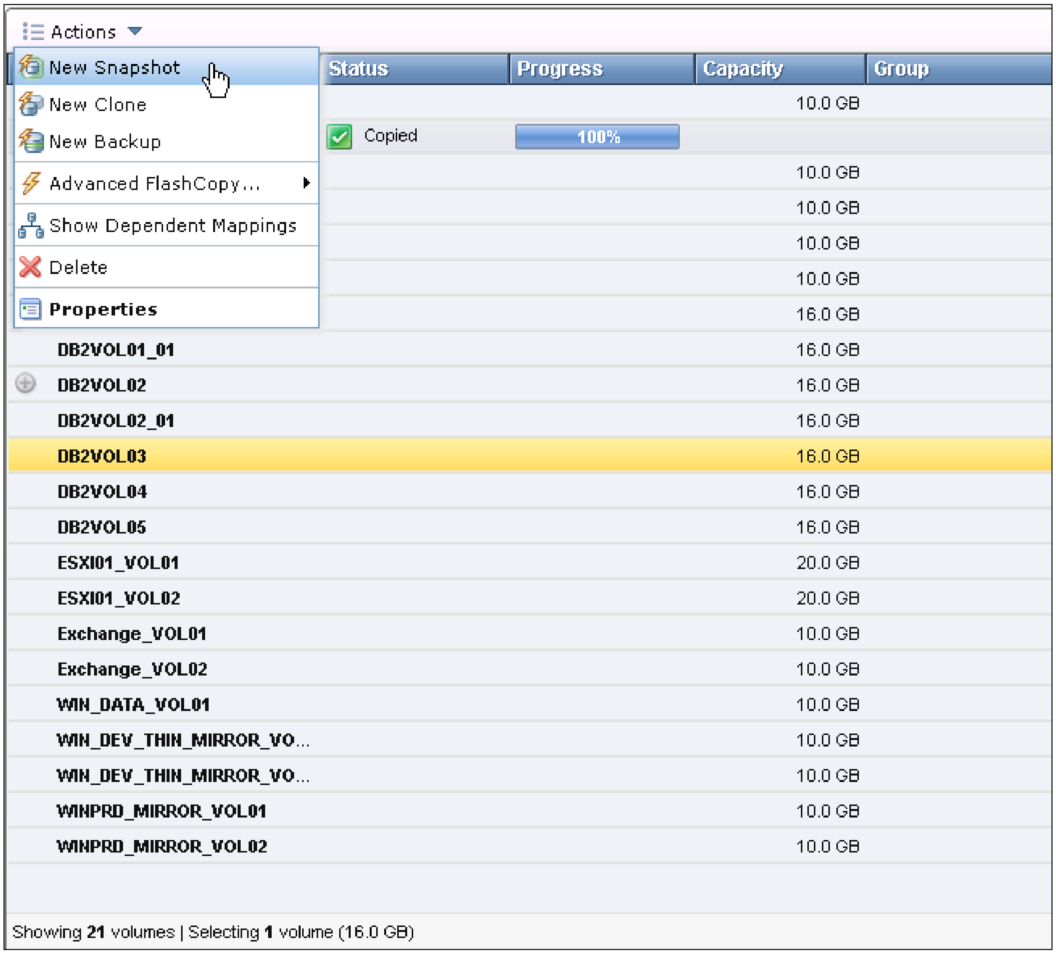


Figure 6-28 New Snapshot option

4. A volume is created as a target volume for this snapshot in the same pool as the source volume. The FlashCopy mapping is created, and it is started.

   You can check the FlashCopy progress in the Progress column Status area (Figure 6-29 on page 249).

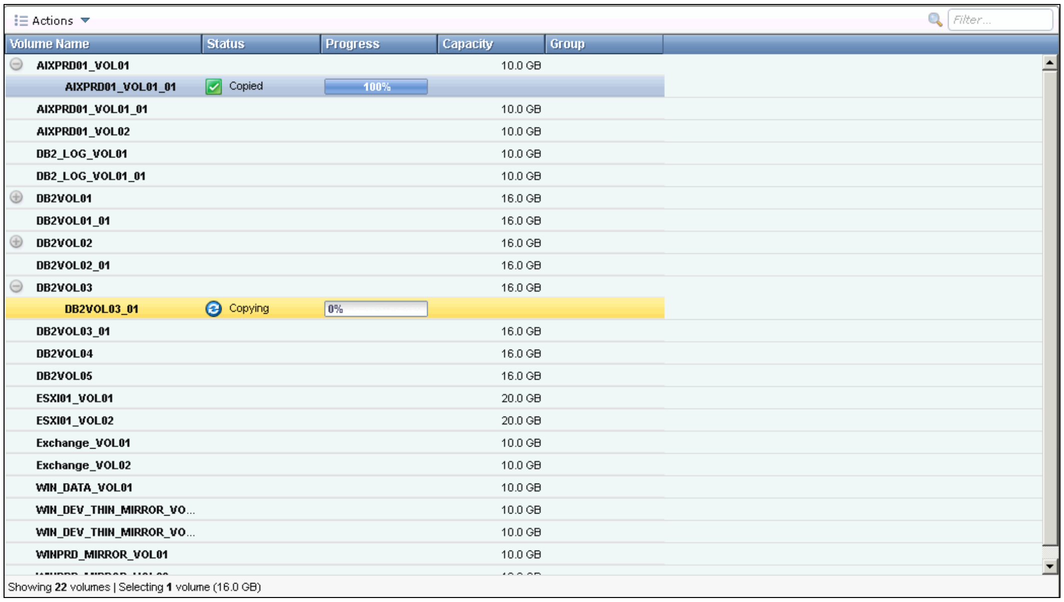


Figure 6-29 Snapshot created and started

6.4.4 Creating and starting a clone preset with a single click

Clone preset: The clone preset creates an exact replica of the volume, which can be changed without affecting the original volume. After the copy completes, the mapping that was created by the preset is automatically deleted.

Clone preset parameters:

- Background copy rate: 50
- Incremental: No
- Delete after completion: Yes
- Cleaning rate: 50
- The target pool is the primary copy source pool.

To create and start a clone with one click, perform these steps:

1. From the SVC/V7000 Welcome panel, click **Copy Services** in the left menu and then click **FlashCopy**.
2. Select the volume that you want to clone.

3. Click **Actions** → **New Clone** (Figure 6-30 on page 250).

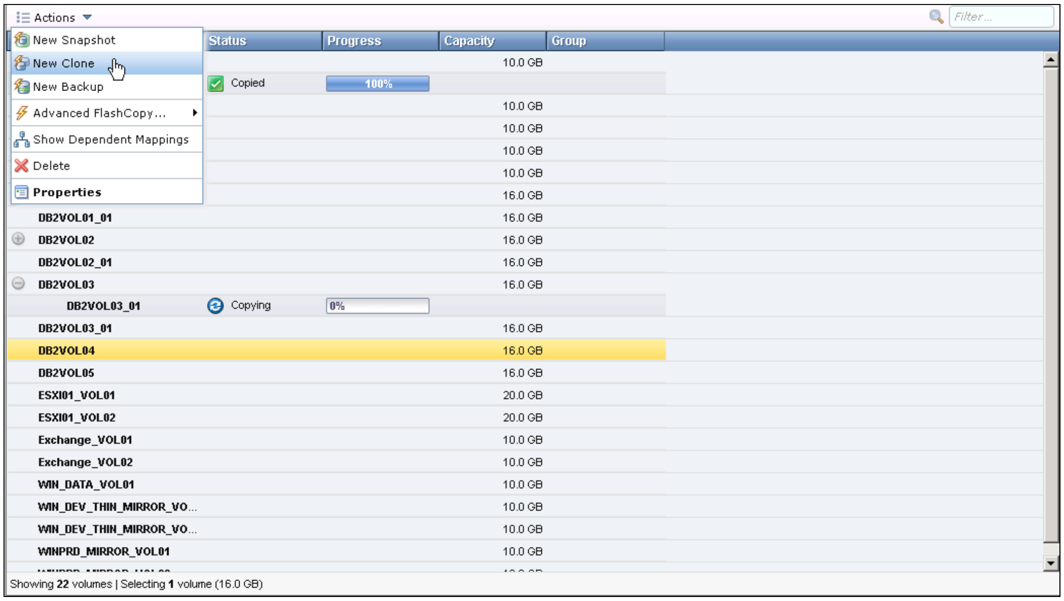


Figure 6-30 New Clone option

4. A volume is created as a target volume for this clone in the same pool as the source volume. The FlashCopy mapping is created and started (Figure 6-31 on page 251).

   You can check the FlashCopy progress in the Progress column or in the Running Tasks Status column.

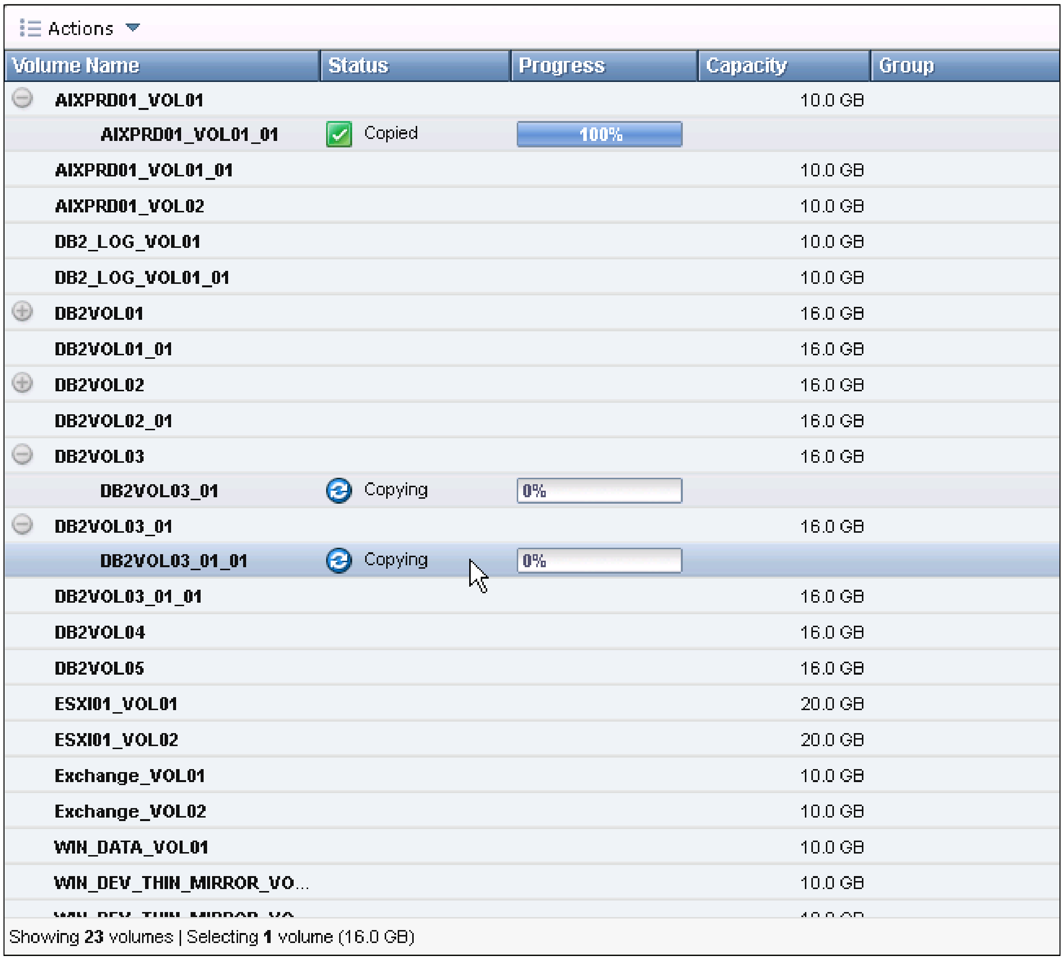


Figure 6-31 Clone created and started

6.4.5 Creating and starting a backup preset with a single click

Backup preset: The backup preset creates a point-in-time replica of the production data. After the copy completes, the backup view can be refreshed from the production data, with minimal copying of data from the production volume to the backup volume.

Backup preset parameters:

- Background Copy rate: 50
- Incremental: Yes
- Delete after completion: No
- Cleaning rate: 50
- The target pool is the primary copy source pool.

To create and start a backup with one click, perform these steps:

1. From the SVC/V7000 Welcome panel, click **Copy Services** in the left menu and then click **FlashCopy**.
2. Select the volume that you want to back up.
3. Click **Actions** → **New Backup** (Figure 6-32 on page 252).

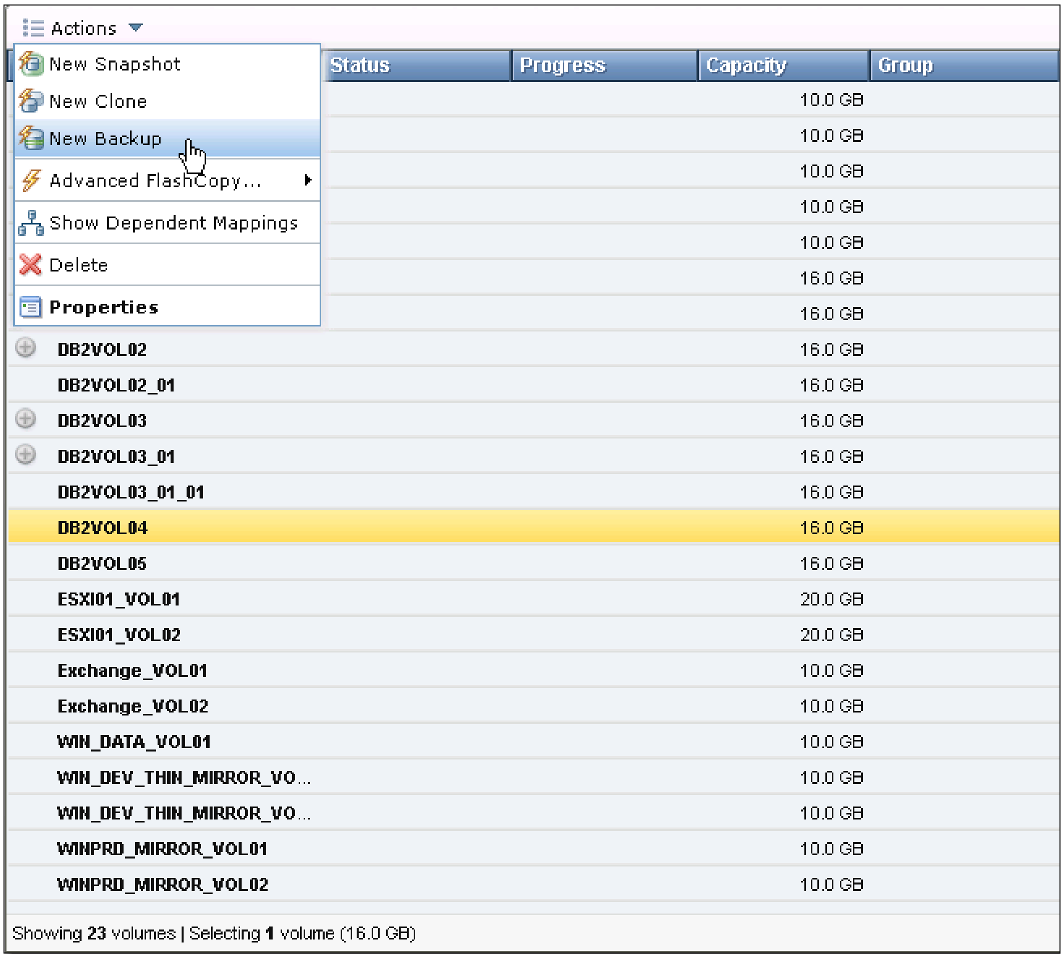


Figure 6-32 New Backup option

4. A volume is created as a target volume for this backup in the same pool as the source volume. The FlashCopy mapping is created and started.

   You can check the FlashCopy progress in the Progress column or in the Running Tasks Status column (Figure 6-33 on page 253).

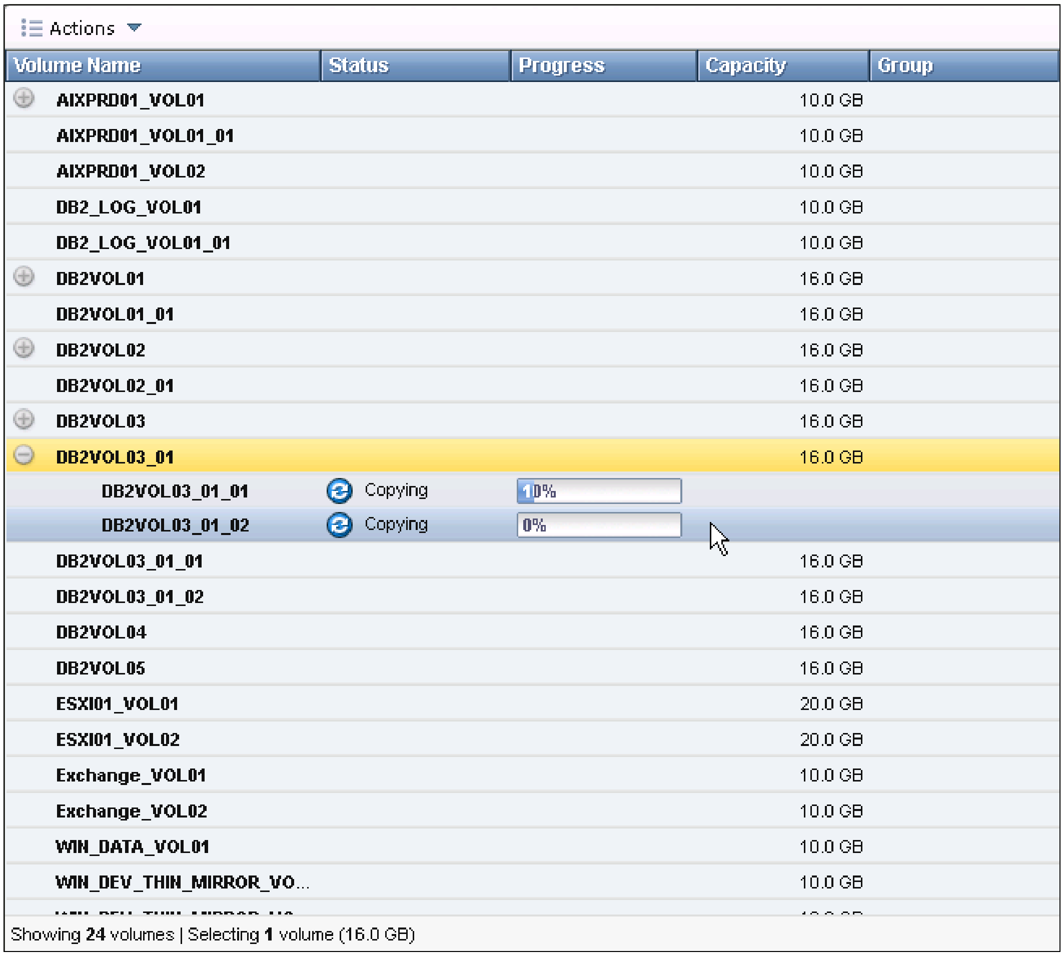


Figure 6-33 Backup created and started

6.4.6 Creating a FlashCopy Consistency Group

To create a FlashCopy Consistency Group in the SVC/V7000 GUI, perform these steps:

1. From the SVC/V7000 Welcome panel, click **Copy Services** and then click **Consistency Groups**. The Consistency Groups panel opens (Figure 6-34 on page 254).

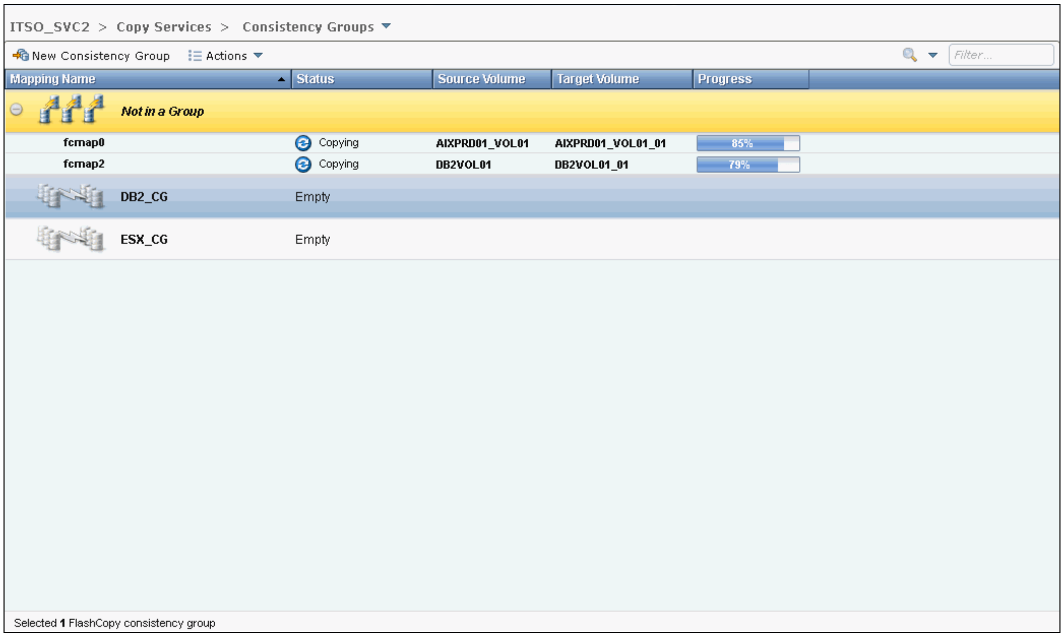


Figure 6-34 Consistency Group panel

2. Click **New Consistency Group** (Figure 6-35).

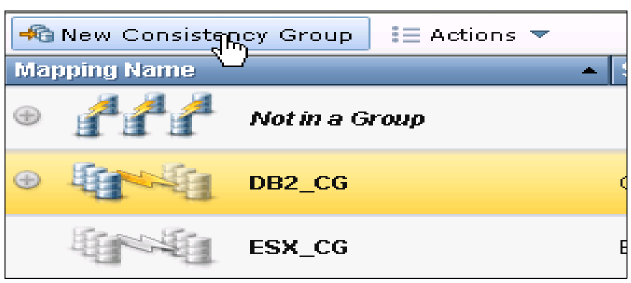


Figure 6-35 Create a FlashCopy Consistency Group

3. Enter the desired FlashCopy Consistency Group name and click **Create** (Figure 6-36).

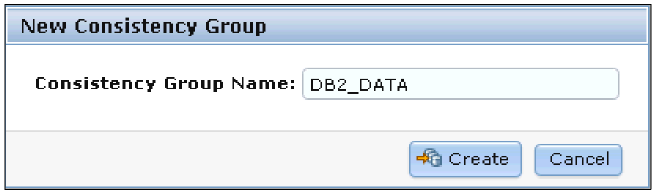


Figure 6-36 New Consistency Group window

> **Consistency Group name:** You can use the letters A to Z and a to z, the numbers 0 to 9, and the underscore (_) character. The volume name can be between one and 63 characters in length.

4. Figure 6-37 on page 255 shows the result.

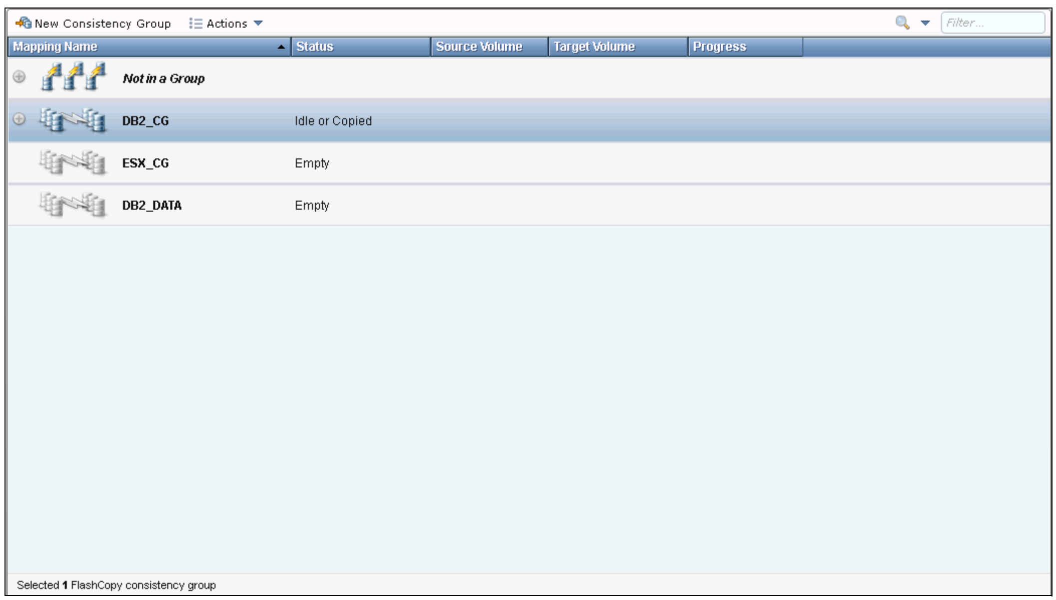


Figure 6-37 View Consistency Group

6.4.7 Creating FlashCopy mappings in a Consistency Group

In this section, we demonstrate how to create FlashCopy mappings for volumes with their respective targets. The source and target volumes were created prior to this operation.

To perform this action, follow these steps:

1. From the SVC/V7000 Welcome panel, click **Copy Services** and then click **Consistency Groups**. The Consistency Groups panel opens (see Figure 6-34 on page 254).

2. Select in which Consistency Group (Figure 6-38) you want to create the FlashCopy mapping.

   If you prefer not to create a FlashCopy mapping in a Consistency Group, select **Not in a Group** in the list.

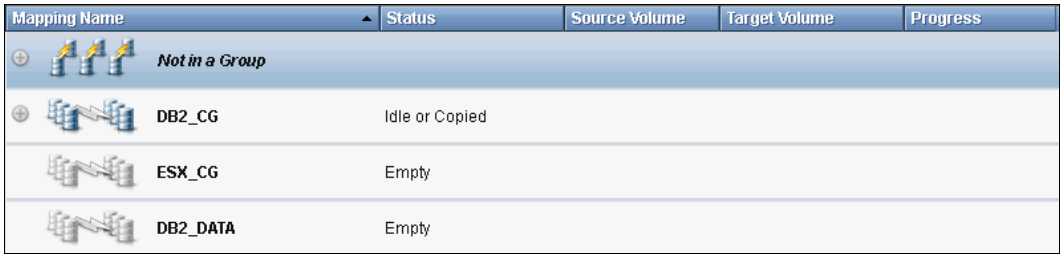


Figure 6-38 Consistency Group selection

3. If you select a Consistency Group, click **Actions** → **New FlashCopy Mapping** (Figure 6-39 on page 256).

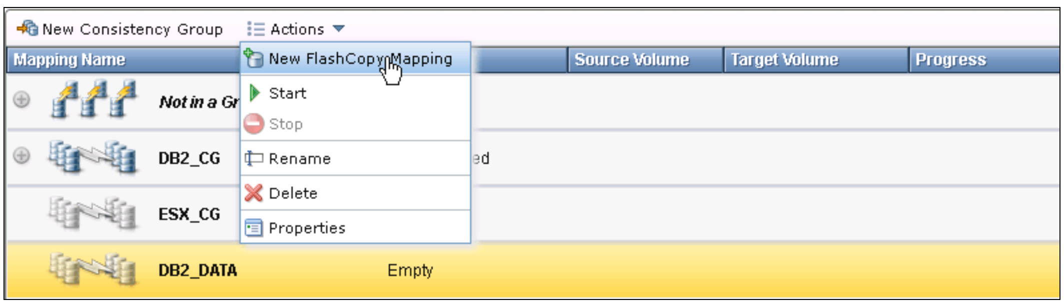


Figure 6-39 New FlashCopy Mapping action for a Consistency Group

4. If you did not select a Consistency Group, click **New FlashCopy Mapping** (Figure 6-40).

Consistency Group consideration: If no Consistency Group is defined, the mapping is a stand-alone mapping. It can be prepared and started without affecting other mappings. All mappings in the same Consistency Group must have the same status to maintain the "consistency" of the group.

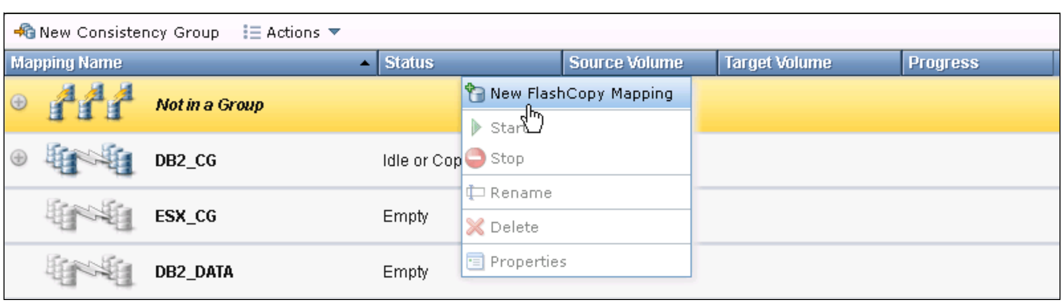


Figure 6-40 New FlashCopy Mapping

5. The New FlashCopy Mapping window opens (Figure 6-41). In this window, you must create the relationships between the source volumes (the disks that are copied) and the target volumes (the disks that receive the copy). A mapping can be created between any two volumes in a cluster.

Important: The source and target volumes must be of equal size.

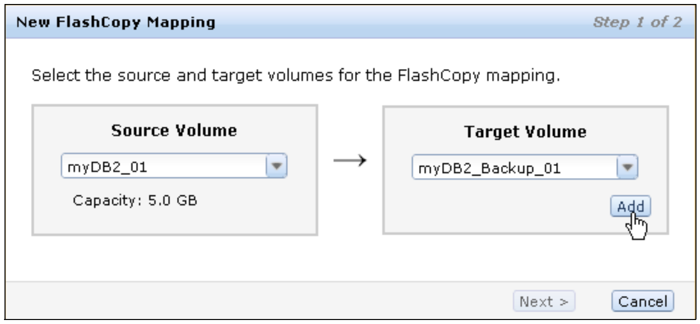


Figure 6-41 New FlashCopy Mapping

Tip: The volumes do not have to be in the same I/O Group or storage pool.

6. Select a volume in the Source Volume column using the drop-down list box. Then, select a volume in the Target Volume column using the drop-down list box. Click **Add**, as shown in Figure 6-41 on page 257. Repeat this action to create other relationships.

   To remove a relationship that has been created, use .

Important: The source volume and target volume must be of equal size. So for a given source volume, only the targets with the appropriate size are shown.

7. Click **Next** after all the relationships that you want to create are shown (Figure 6-42).

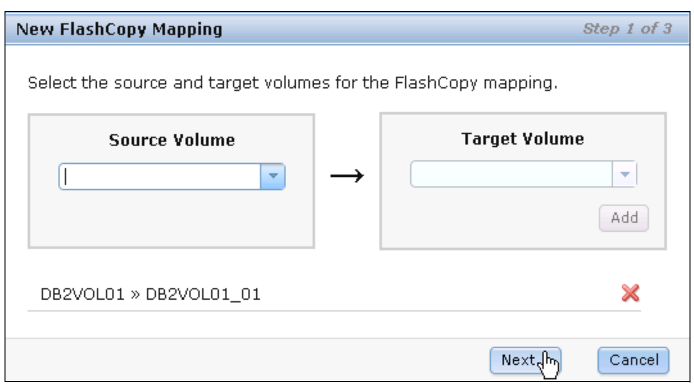


Figure 6-42 New FlashCopy Mapping with the relationships that have been created

8. In the next window, you need to select one FlashCopy preset. The GUI interface provides three presets (Snapshot, Clone, or Backup) to simplify the more common FlashCopy operations (Figure 6-43).

   The presets and their use cases are described here:

   Snapshot This preset creates a copy-on-write point-in-time copy.

   Clone This preset creates an exact replica of the source volume on a target volume. The copy can be changed without affecting the original volume.

Backup This preset creates a FlashCopy mapping that can be used to recover data or objects if the system experiences data loss. These backups can be copied multiple times from the source and target volumes.

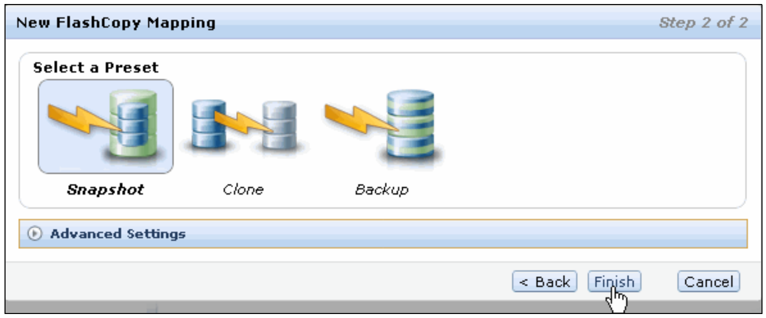


Figure 6-43 New FlashCopy Mapping window

You can customize various advanced options for whichever preset you choose. To access these settings, click **Advanced Settings**.

If you prefer not to customize these settings, go directly to step 9.

You can customize the following options, as shown in Figure 6-44:

- Background Copy Rate: This option determines the priority that is given to the copy process. A faster rate increases the priority of the process, which might affect the performance of other operations.
- Incremental: This option copies only the parts of the source or target volumes that have changed since the last copy. Incremental copies reduce the completion time of the copy operation.

  Incremental copies: Even if the type of the FlashCopy mapping is incremental, the first copy process copies all of the data from the source to the target volume.

- Delete after completion: This option automatically deletes a FlashCopy mapping after the background copy is completed. Do not use this option when the background copy rate is set to zero (0).
- Cleaning Rate: This option minimizes the amount of time that a mapping is in the stopping state. If the mapping has not completed, the target volume is offline while the mapping is stopping.

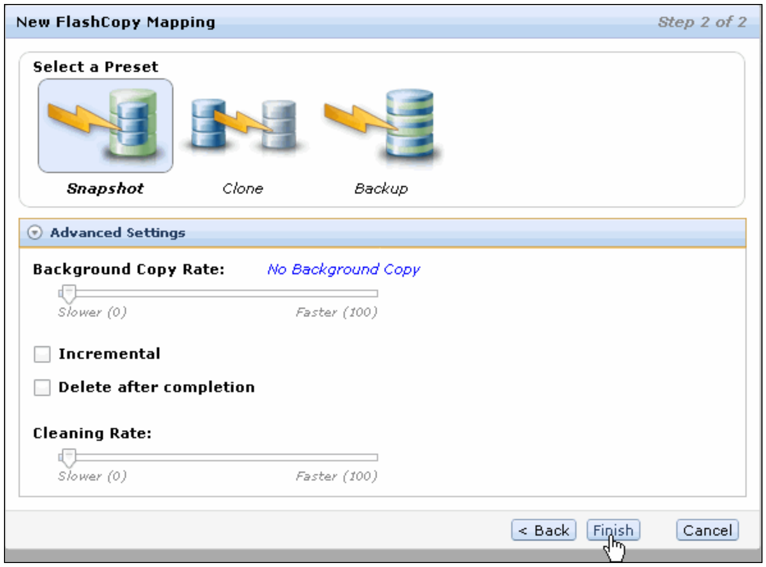


Figure 6-44 New FlashCopy Mapping Advanced Settings

9. If you did not create these FlashCopy mappings from a Consistency Group (see step 3 on page 256), you must confirm your choice by selecting **No, do not add the mappings to a consistency group** (Figure 6-45).

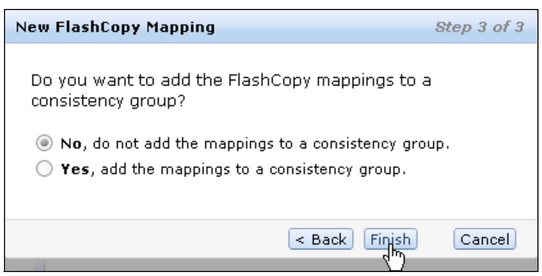


Figure 6-45 Do not add the mappings to a consistency group

10. Click **Finish**, as shown in Figure 6-44 on page 259.

11.Check the result of this FlashCopy mapping in the Consistency Groups window (Figure 6-46).

For each FlashCopy mapping relationship that you have created, a mapping name is automatically generated starting with `fcmapX` where `X` is an available number.

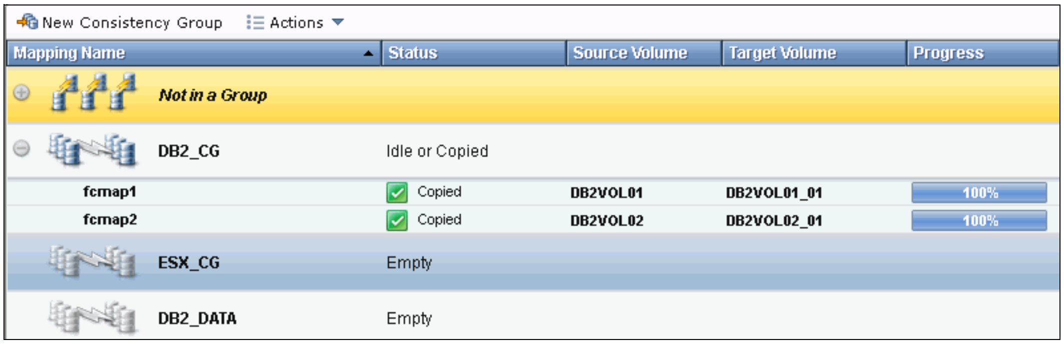


Figure 6-46 FlashCopy mappings result

> **Tip:** You can invoke FlashCopy from the SVC GUI, but using the SVC GUI might be impractical if you plan to handle a large number of FlashCopy mappings or Consistency Groups periodically, or at varying times. In this case, creating a script by using the CLI might be more convenient.

6.4.8 Showing dependent mappings

Perform the following steps to show dependent mappings for a given FlashCopy mapping:

1. From the SVC/V7000 Overview panel, click **Copy Services** in the left menu and then click either the FlashCopy, Consistency Groups, or FlashCopy Mappings panel.
2. Select the volume (from the FlashCopy panel only) or the FlashCopy mapping that you want to remove from a Consistency Group.
3. Click **Actions** → **Show Dependent Mappings** (Figure 6-47).

> **Tip:** You can also right-click a FlashCopy mapping and select **Show Dependent Mappings** from the list.

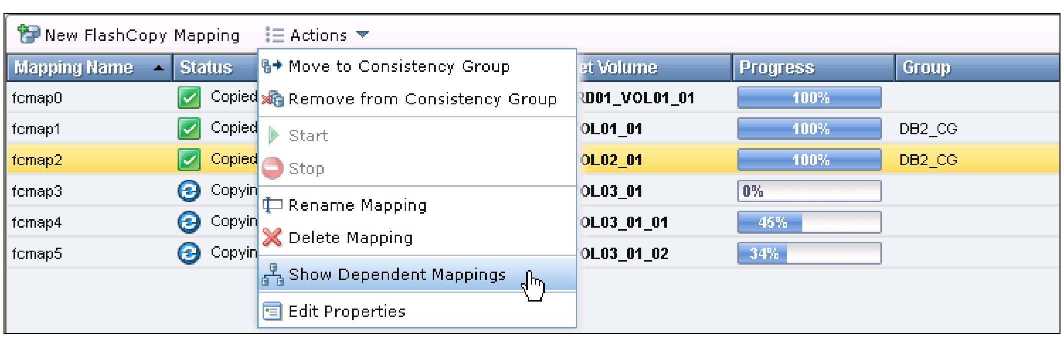


Figure 6-47 Show Dependent Mappings

In the Dependent Mappings window (Figure 6-48), you can see the dependent mapping for a given volume or a FlashCopy mapping. If you click one of these volumes, you can see its properties.

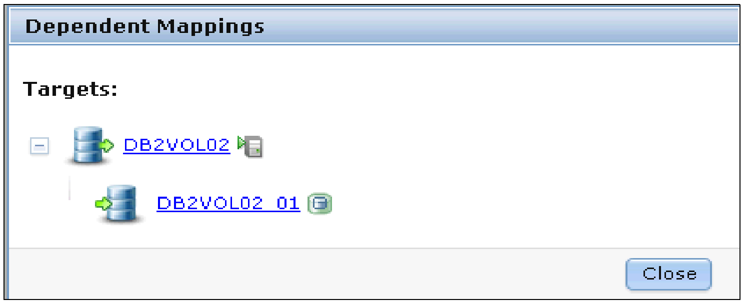


Figure 6-48 Dependent Mappings

4. Click **Close** to close this window.

6.4.9 Moving a FlashCopy mapping to a Consistency Group

Perform the following steps to move a FlashCopy mapping to a Consistency Group:

1. From the SVC/V7000 Welcome panel, click **Copy Services** in the left menu and then click either the FlashCopy, Consistency Groups, or FlashCopy Mappings panel.
2. Select the FlashCopy mapping that you want to move to a Consistency Group or the FlashCopy mapping for which you want to change the Consistency Group.
3. Click **Actions** → **Move to Consistency Group** (Figure 6-49).

> **Tip:** You can also right-click a FlashCopy mapping and select **Move to Consistency Group** from the list.

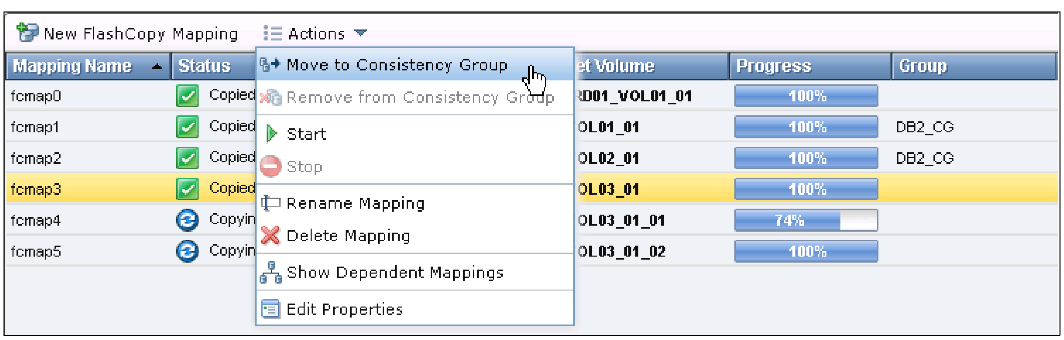


Figure 6-49 Move to Consistency Group action

4. In the Move FlashCopy Mapping to Consistency Group window, select the Consistency Group for this FlashCopy mapping by using the drop-down list box (Figure 6-50 on page 262).

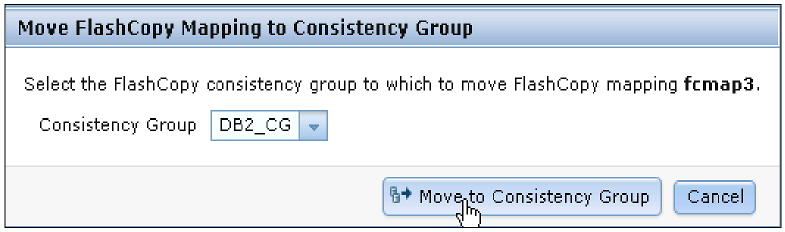


Figure 6-50 Move a FlashCopy mapping to a Consistency Group

5. Click **Move to Consistency Group** to confirm your changes.

6.4.10 Removing a FlashCopy mapping from a Consistency Group

Perform the following steps to remove a FlashCopy mapping from a Consistency Group:

1. From the SVC/V7000 Overview panel, click **Copy Services** in the left menu and then click either the FlashCopy, Consistency Groups, or FlashCopy Mappings panel.
2. Select the FlashCopy mapping that you want to remove from a Consistency Group**.**
3. Click **Actions** → **Remove from Consistency Group** (Figure 6-51).

> **Tip:** You can also right-click a FlashCopy mapping and select **Remove from Consistency Group** from the list.

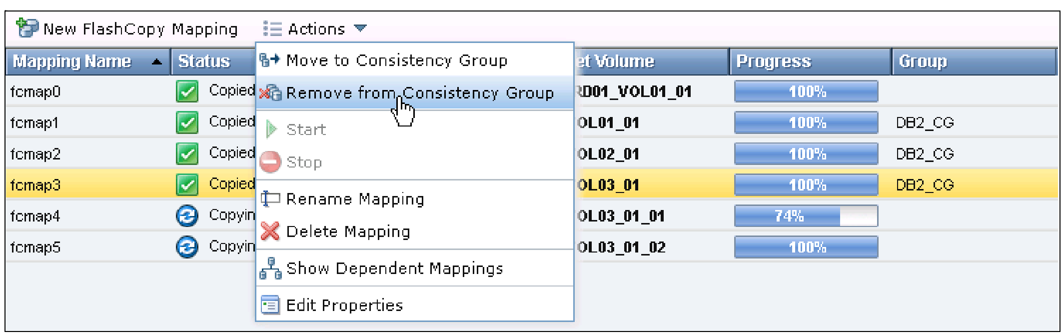


Figure 6-51 Remove from Consistency Group action

In the Remove FlashCopy Mapping from Consistency Group window, click **Remove** (Figure 6-52).

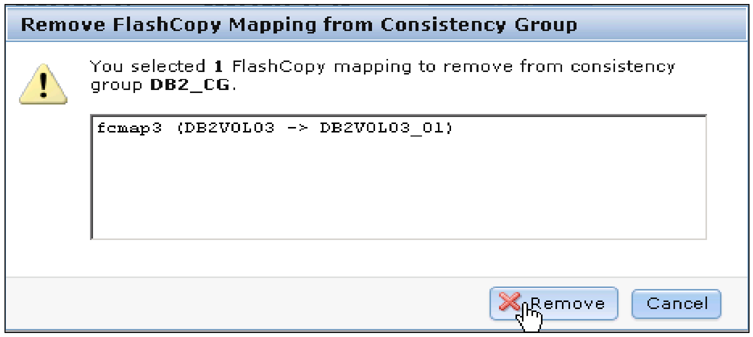


Figure 6-52 Remove FlashCopy Mapping from Consistency Group

6.4.11 Modifying a FlashCopy mapping

Perform the following steps to modify a FlashCopy mapping:

1. From the SVC/V7000 Welcome panel, click **Copy Services** in the left menu and then click either the FlashCopy, Consistency Groups, or FlashCopy Mappings panel.
2. Select the FlashCopy mapping that you want to modify in the table.
3. Click **Actions** → **Edit Properties** (Figure 6-53).

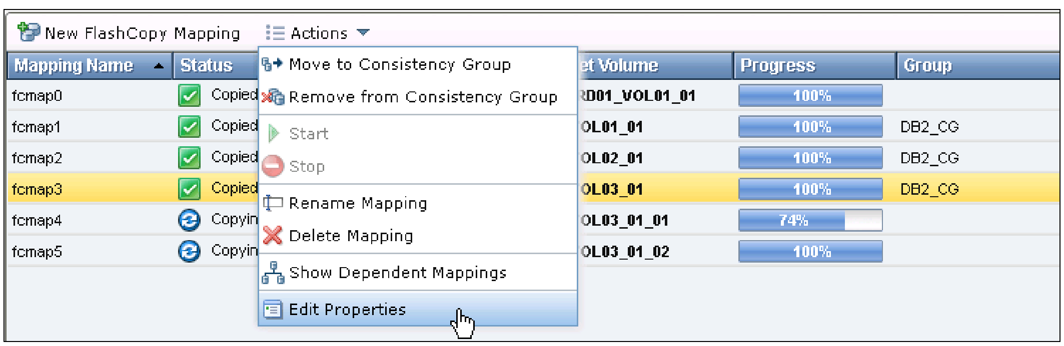


Figure 6-53 Edit Properties

> **Tip:** You can also right-click a FlashCopy mapping and select **Edit Properties** from the list.

4. In the Edit FlashCopy Mapping window, you can modify the following parameters for a selected FlashCopy mapping (Figure 6-54):
   - Background Copy Rate: This option determines the priority that is given to the copy process. A faster rate increases the priority of the process, which might affect the performance of other operations.
   - Cleaning Rate: This option minimizes the amount of time that a mapping is in the stopping state. If the mapping has not completed, the target volume is offline while the mapping is stopping.

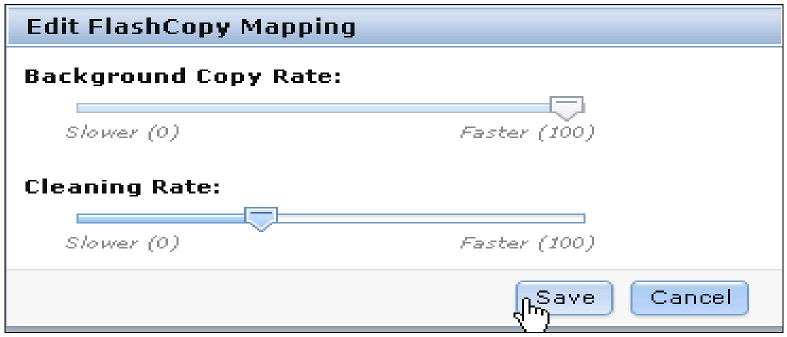


Figure 6-54 Edit FlashCopy Mapping

5. Click **Save** to confirm your changes.

6.4.12 Renaming a FlashCopy mapping

Perform the following steps to rename a FlashCopy mapping:

1. From the SVC/V7000 Welcome panel, click **Copy Services** and then click either Consistency Groups or FlashCopy Mappings.
2. In the table, select the FlashCopy mapping that you want to rename.
3. Click **Actions** → **Rename Mapping** (Figure 6-55).

> **Tip:** You can also right-click a FlashCopy mapping and select **Rename** from the list.

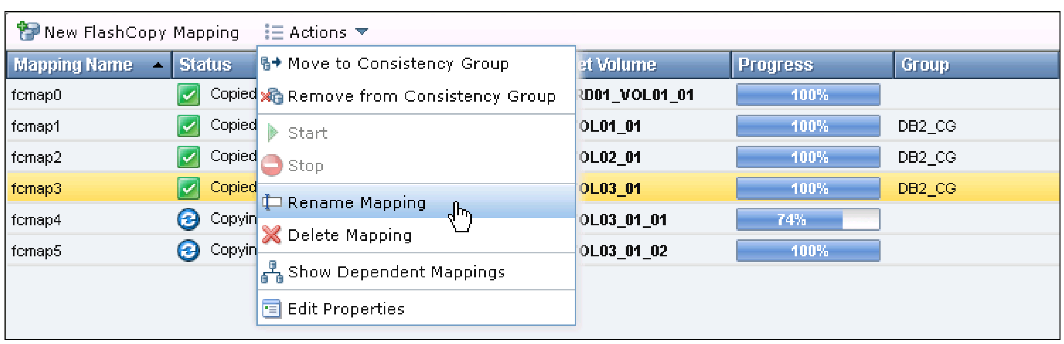


Figure 6-55 Rename Mapping action

4. In the Rename Mapping window, type the new name that you want to assign to the FlashCopy mapping and click **Rename** (Figure 6-56).

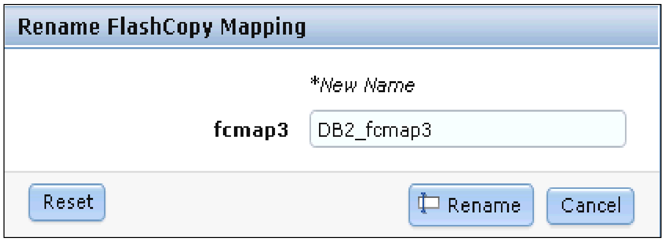


Figure 6-56 Renaming a FlashCopy mapping

> **FlashCopy name:** You can use the letters A to Z and a to z, the numbers 0 to 9, and the underscore (_) character. The mapping name can be between one and 63 characters in length.

6.4.13 Renaming a Consistency Group

To rename a Consistency Group, perform the following steps:

1. From the SVC/V7000 Overview panel, click **Copy Services menu** and then click **Consistency Group**.
2. From the left panel, select the Consistency Group that you want to rename. Then, select **Actions** → **Rename** (Figure 6-57 on page 265).

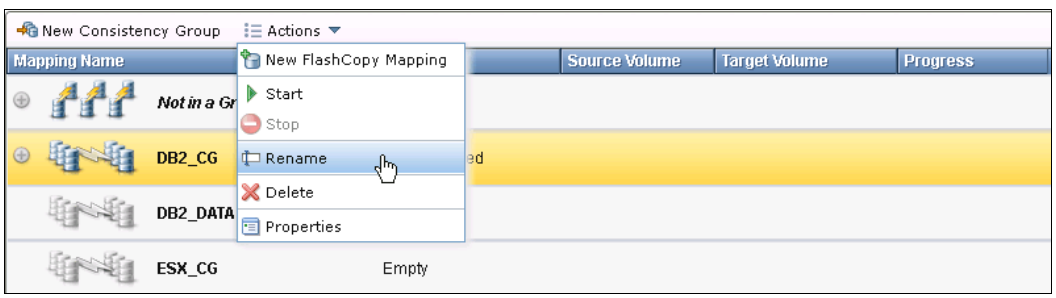


Figure 6-57 Renaming a Consistency Group

3. Type the new name that you want to assign to the Consistency Group and click **Rename** (Figure 6-58).

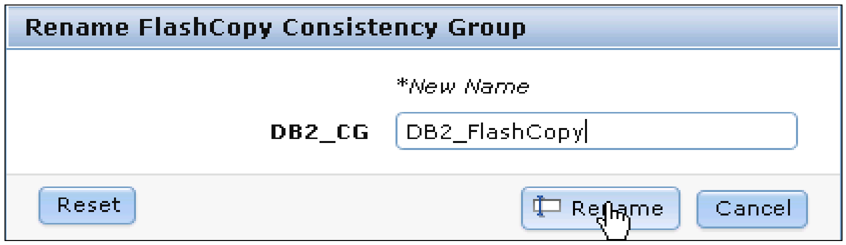


Figure 6-58 Changing the name for a Consistency Group

> **Consistency Group name:** The name can consist of the letters A to Z and a to z, the numbers 0 to 9, the dash (-), and the underscore (_) character. The name can be between one and 63 characters in length. However, the name cannot start with a number, the dash or the underscore.

4. From the Consistency Group panel, the new Consistency Group name is displayed.

6.4.14 Deleting a FlashCopy mapping

Perform the following steps to delete a FlashCopy mapping:

1. From the SVC/V7000 Overview panel, click **Copy Services** and then click the FlashCopy, Consistency Groups, or FlashCopy Mappings icon.
2. In the table, select the FlashCopy mapping that you want to delete.

> **Selecting multiple FlashCopy mappings:** To select multiple FlashCopy mappings, hold down the **Ctrl** key and use the mouse to select the entries.
>
> This capability is only available in the Consistency Groups and FlashCopy Mappings panels.

3. Click **Actions** → **Delete Mapping** (Figure 6-59 on page 266).

> **Tip:** You can also right-click a FlashCopy mapping and select **Delete Mapping** from the list.

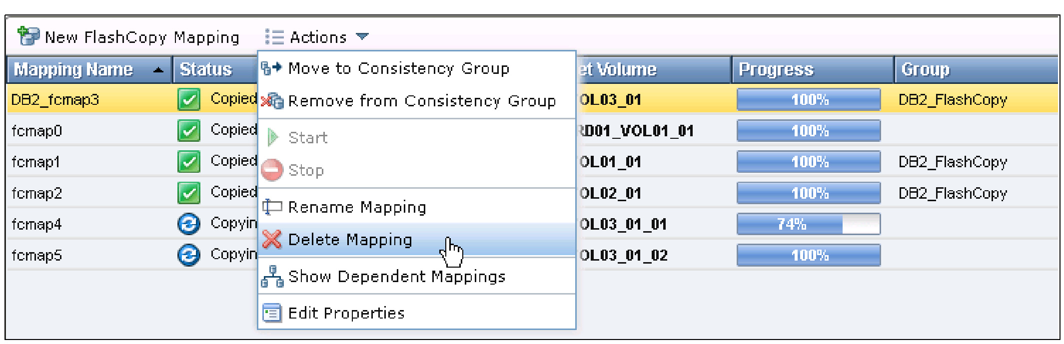


Figure 6-59 Delete Mapping action

4. The Delete FlashCopy Mapping window opens, as shown in Figure 6-60. In the “Verify the number of FlashCopy mappings that you are deleting” field, you need to enter the number of volumes that you want to remove. This verification has been added to help you to avoid deleting the wrong mappings.

   If you still have target volumes that are inconsistent with the source volumes and you definitely want to delete these FlashCopy mappings, select **Delete the FlashCopy mapping even when the data on the target volume is inconsistent, or if the target volume has other dependencies**.

   Click **Delete** to complete the operation (Figure 6-60).

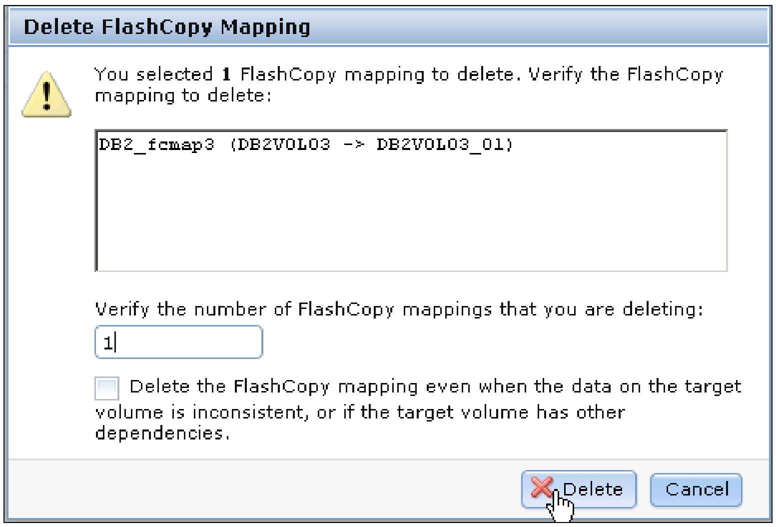


Figure 6-60 Delete FlashCopy Mapping

6.4.15 Deleting a FlashCopy Consistency Group

Important: Deleting a Consistency Group does not delete the FlashCopy mappings.

Perform the following steps to delete a FlashCopy Consistency Group:

1. From the SVC/V7000 Overview panel, click **Copy Services** and then click the **Consistency Groups** panel.
2. Select the FlashCopy Consistency Group that you want to delete.
3. Click **Actions** → **Delete** (Figure 6-61 on page 267).

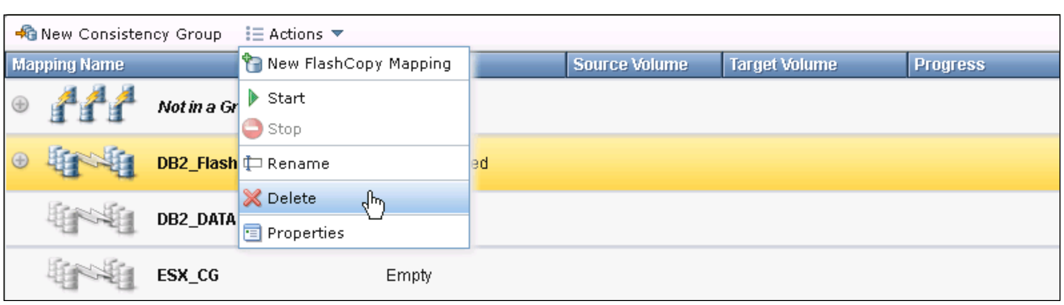


Figure 6-61 Delete action

4. The Warning window opens (Figure 6-62). Click **OK** to complete the operation.

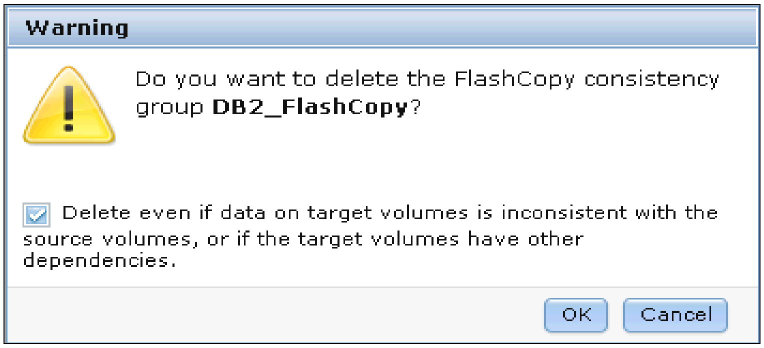


Figure 6-62 Warning window

6.4.16 Starting FlashCopy mappings

When the FlashCopy mapping is created, the copy process can be started. Only mappings that are not a member of a Consistency Group, or the only mapping in a Consistency Group, can be started individually. Follow these steps:

1. From the SVC/V7000 Welcome panel, click **Copy Services** and then click either the FlashCopy or the FlashCopy Mappings panel.
2. Select the FlashCopy mapping that you want to start in the table.
3. Click **Actions** → **Start** (Figure 6-63) to start the FlashCopy Mapping.

Tip: You can also right-click a FlashCopy mapping and select **Start** from the list.

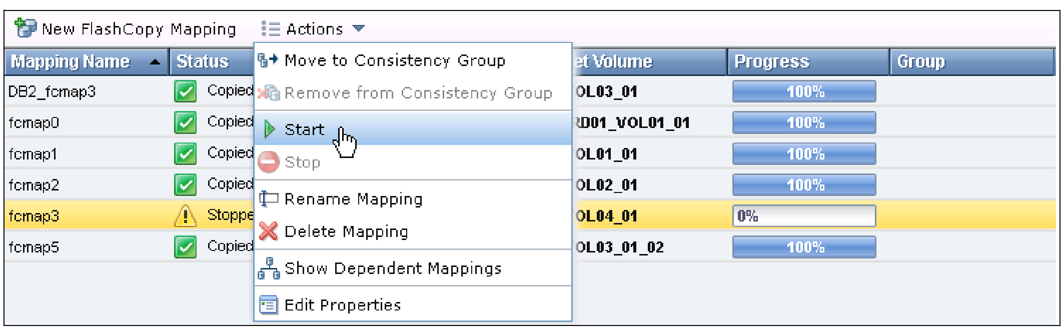


Figure 6-63 Start action

4. You can check the FlashCopy progress in the Progress column of the table (Figure 6-64) or in the Running Tasks status area.

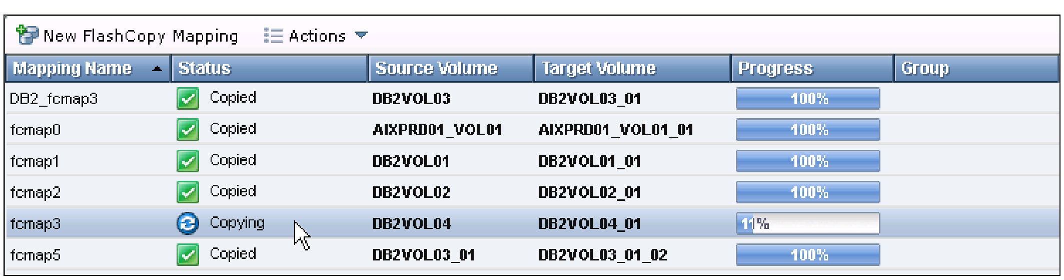


Figure 6-64 Checking FlashCopy progress

5. After the task completes, the FlashCopy status is in a `Copied` state (Figure 6-65).

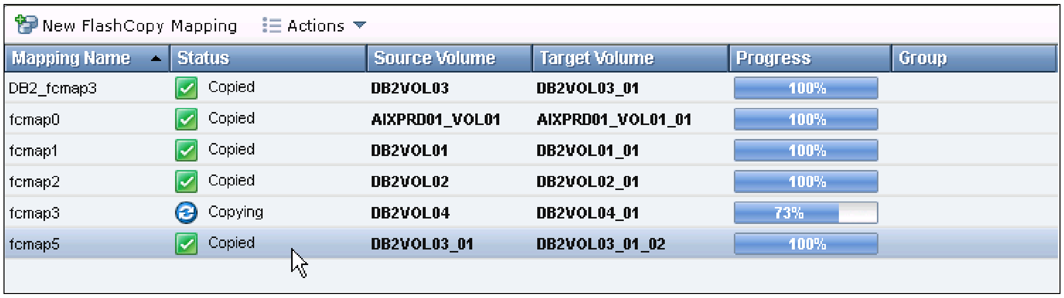


Figure 6-65 Example of copied FlashCopy

6.4.17 Starting a FlashCopy Consistency Group

All of the mappings in a Consistency Group will be brought to the same state. To start the FlashCopy Consistency Group, perform these steps:

1. From the SVC/V7000 Overview window, click **Copy Services** and then click the **Consistency Groups** panel.
2. Select the Consistency Group that you want to start (Figure 6-66).

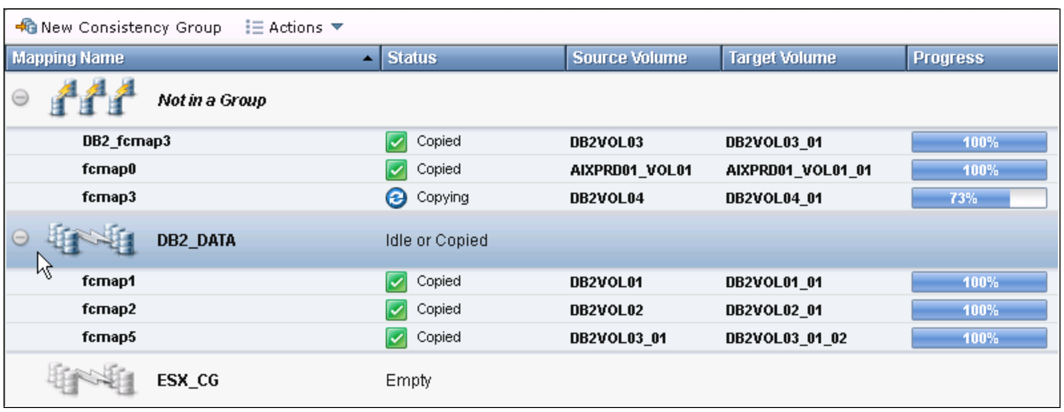


Figure 6-66 FlashCopy Consistency Groups window

3. Click **Actions** → **Start** (Figure 6-67 on page 269) to start the FlashCopy Consistency Group.

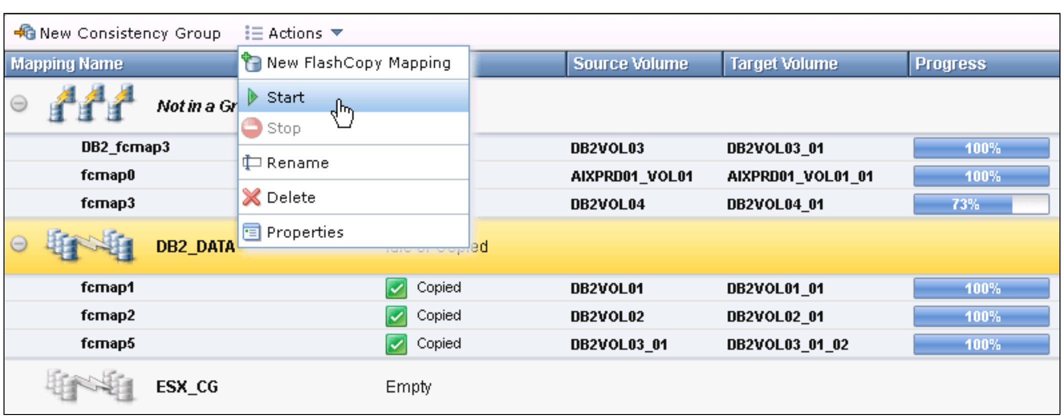


Figure 6-67 Start action

4. You can check the FlashCopy Consistency Group progress in the Progress column (Figure 6-68) or in the Running Tasks status area.

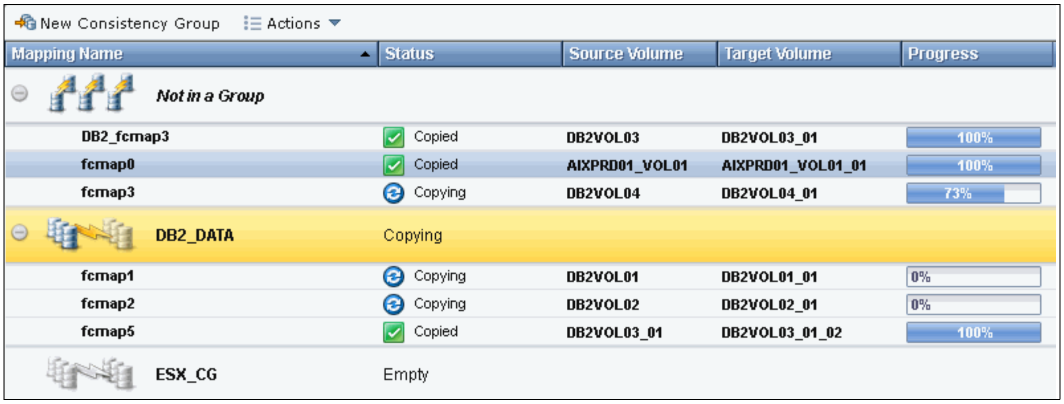


Figure 6-68 Checking FlashCopy Consistency Group progress

5. After the task completes, the FlashCopy status is in a `Copied` state (Figure 6-69).

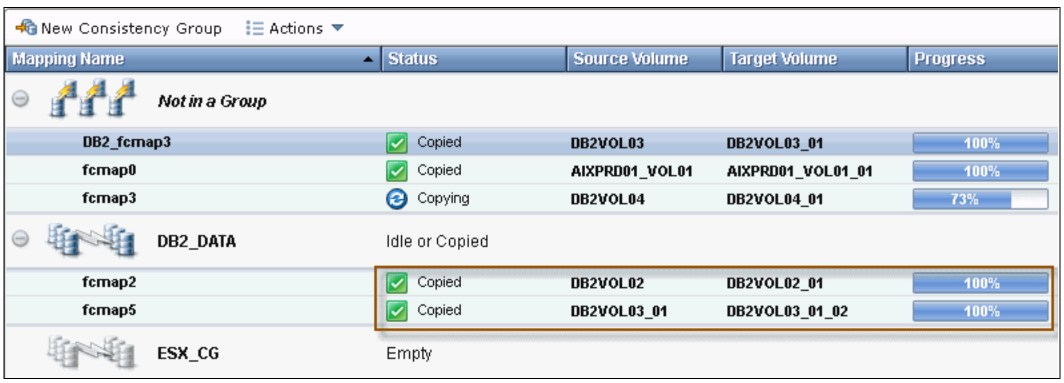


Figure 6-69 Copied FlashCopy Consistency Group

6.4.18 Stopping the FlashCopy Consistency Group

When a FlashCopy Consistency Group is stopped, the target volumes become invalid and are set offline by the cluster. The FlashCopy mapping or Consistency Group must be prepared again or retriggered to bring the target volumes online again.

> **Important:** Only stop a FlashCopy mapping when the data on the target volume is useless, or if you want to modify the FlashCopy mapping.
>
> When a FlashCopy mapping is stopped, the target volume becomes invalid and is set offline by the cluster, as shown in Figure 6-72 on page 271.

Perform the following steps to stop a FlashCopy Consistency Group:

1. From the SVC/V7000 Overview panel, click **Copy Services** and then click either the FlashCopy, Consistency Groups, or the FlashCopy Mappings panel.
1. In the table, select the FlashCopy mapping that you want to stop.

2. Click **Actions** → **Stop** (Figure 6-70) to stop the FlashCopy mapping.

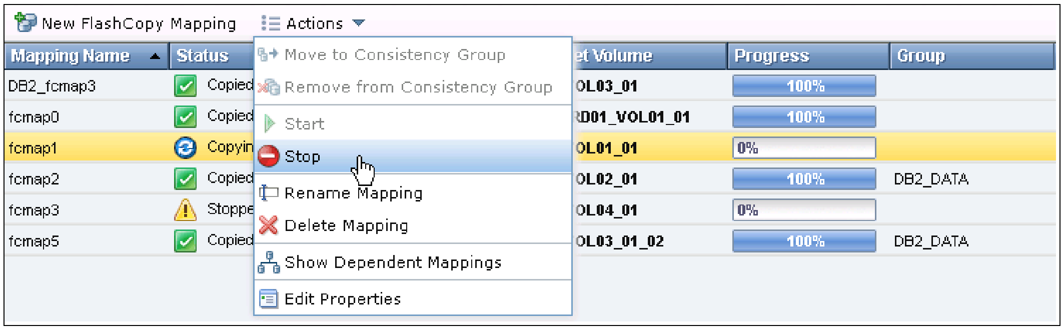


Figure 6-70 Stop action

3. Notice that the FlashCopy mapping status has changed to `Stopped` (Figure 6-71).

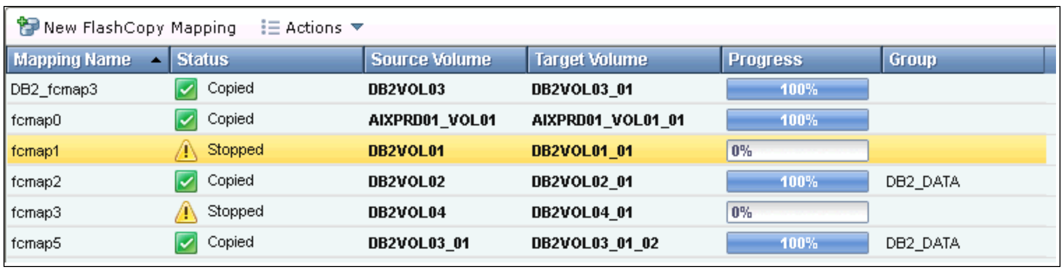


Figure 6-71 FlashCopy Consistency Group status

4. The targeted volume is now shown as `Offline` in the Volumes list (Figure 6-72 on page 271).

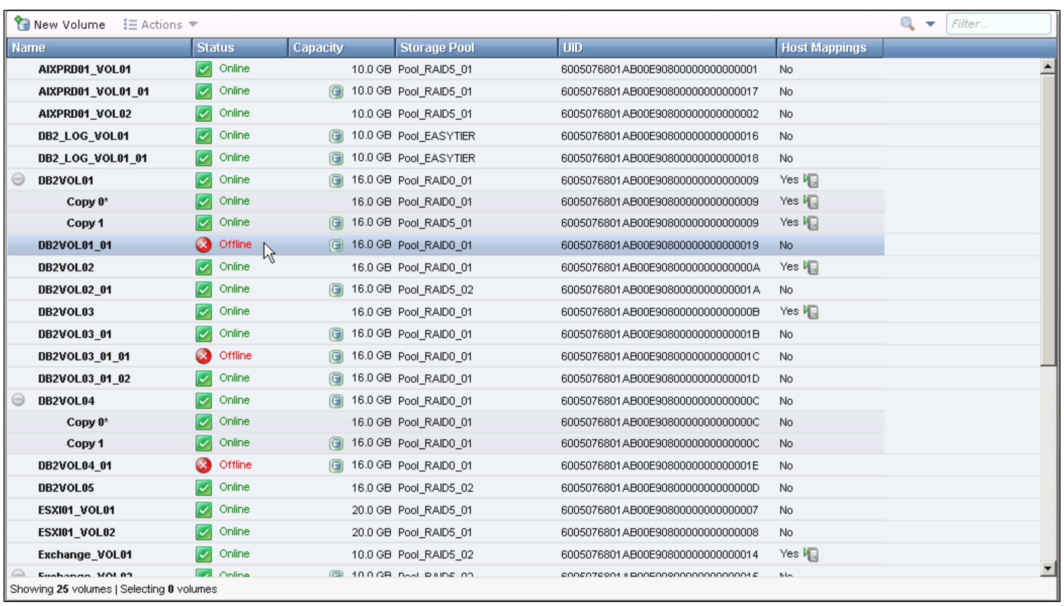


Figure 6-72 Targeted volume is offline

6.4.19 Stopping the FlashCopy mapping

When a FlashCopy is stopped, the target volumes become invalid and are set offline by the cluster. The FlashCopy mapping must be retriggered to bring the target volumes online again.

Important: Only stop a FlashCopy mapping when the data on the target volume is useless, or if you want to modify the FlashCopy mapping.

When a FlashCopy mapping is stopped, the target volume becomes invalid and is set offline by the cluster.

Perform the following steps to stop a FlashCopy mapping:

1. From the SVC/V7000 Welcome panel, click **Copy Services** and then click the **Consistency Groups** panel.
1. In the left side of this panel, select the Consistency Group that you want to stop.
2. Click **Actions** → **Stop** (Figure 6-73) to stop the FlashCopy Consistency Group.

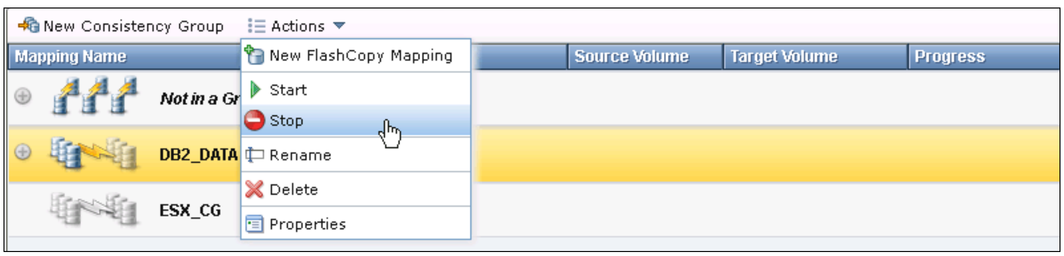


Figure 6-73 Stopping the FlashCopy Consistency Group

3. Notice that the FlashCopy Consistency Group status has now changed to `Stopped` (Figure 6-74 on page 272).

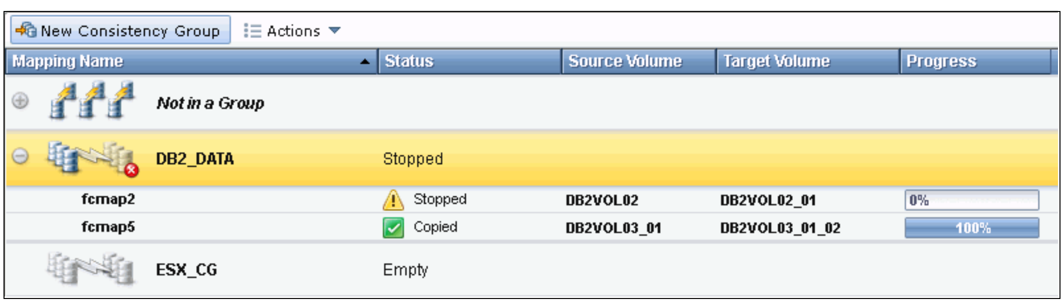


Figure 6-74 FlashCopy Consistency Group status

6.4.20 Migrating between a fully allocated volume and a thin-provisioned volume

If you want to migrate from a fully allocated volume to a thin-provisioned volume, follow the same procedure described in 6.4.2, “Creating a FlashCopy mapping” on page 232. However, make sure that you either select a thin-provisioned volume that has already been created as your target volume, or that you create one. You can use this same method to migrate from a thin-provisioned volume to a fully allocated volume.

Create a FlashCopy mapping with the fully allocated volume as the source and the thin-provisioned volume as the target.

Important: The copy process overwrites all of the data on the target volume. You *must* back up all data before you start the copy process.

6.4.21 Reversing and splitting a FlashCopy mapping

You can now perform a reverse FlashCopy mapping without having to remove the original FlashCopy mapping, and without restarting a FlashCopy mapping from the beginning. Figure 6-75 shows an example of a reverse FlashCopy dependency.

You can start a FlashCopy mapping whose target is the source of another FlashCopy mapping.

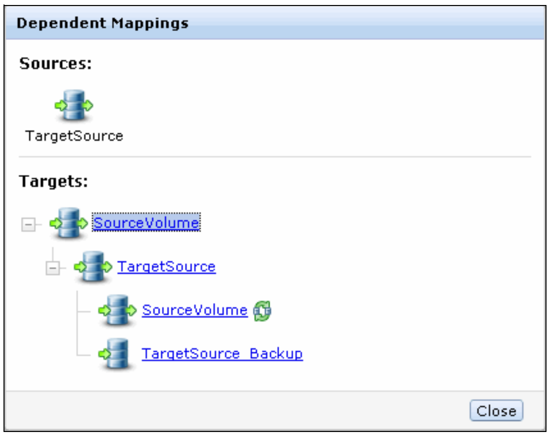


Figure 6-75 Dependent Mappings

This capability enables you to reverse the direction of a FlashCopy map without having to remove existing maps, and without losing the data from the target, (Figure 6-76).

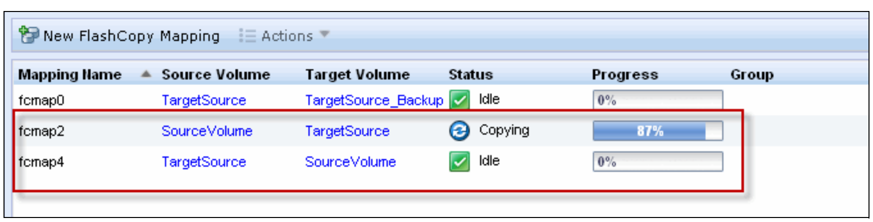


Figure 6-76 Reverse FlashCopy

7

Volume Mirroring

The Volume Mirroring function enables the synchronous and symmetrical mirroring of volumes. An overview of the functionality is provided here, along with descriptions of Mirrored Volume components and Volume Mirroring usage cases and their characteristics. Performance considerations are also highlighted.

The following topics are discussed:

- Volume Mirroring overview
- Volume Mirroring infrastructure design examples

7.1 Volume Mirroring overview

By using Volume Mirroring, you can have two physical copies of a volume providing a basic RAID-1 function. These copies can be in the same Storage Pool or in different Storage Pools, with different extent sizes of the Storage Pool. Typically the two copies are allocated in different Storage Pools.

The first Storage Pool contains the original (primary volume copy). If one storage controller or Storage Pool fails, a volume copy is not affected if it has been placed on a different storage controller or in a different Storage Pool.

If a volume is created with two copies, both copies will use the same virtualization policy. However, there is also a way to have two copies of a volume with different virtualization policies. In combination with *thin-provisioning*, each mirror of a volume can be thin-provisioned or fully allocated and in striped, sequential, or image mode.

A mirrored (secondary) volume has all of the capabilities of the primary volume copy and also the same restrictions (for example, a mirrored volume is owned by an I/O Group, just as any other volume). This feature also provides a *point-in-time copy* functionality that is achieved by "splitting" a copy from the volume. Note, however, that the mirrored volume does not address other forms of mirroring based on Remote Copy (Global or Metro Mirror functions), which mirrors volumes across I/O Groups or clustered systems.

Tip: In the management GUI, an asterisk (*) indicates the primary copy of the mirrored volume, see more details in Figure 7-1.

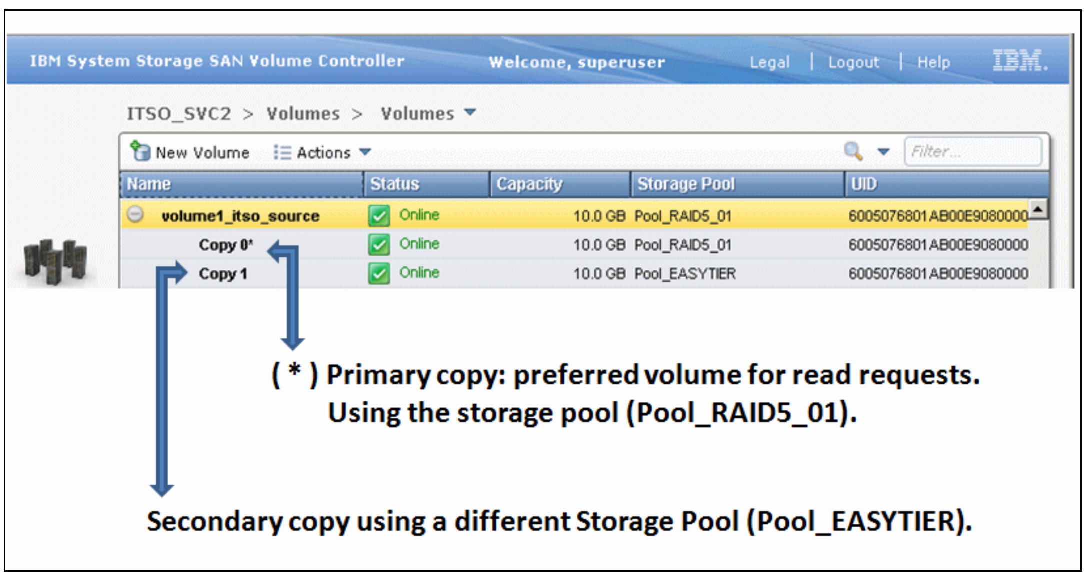


Figure 7-1 Volume Mirroring using different Storage Pools

One copy is the primary copy, and the other copy is the secondary copy. Initially, the first volume copy is the primary copy (in the Storage Pool specified first, Pool_RAID5_01, as illustrated in Figure 7-1). You can change the primary copy to the secondary copy if required.

Figure 7-2 on page 277 provides an overview of Volume Mirroring.

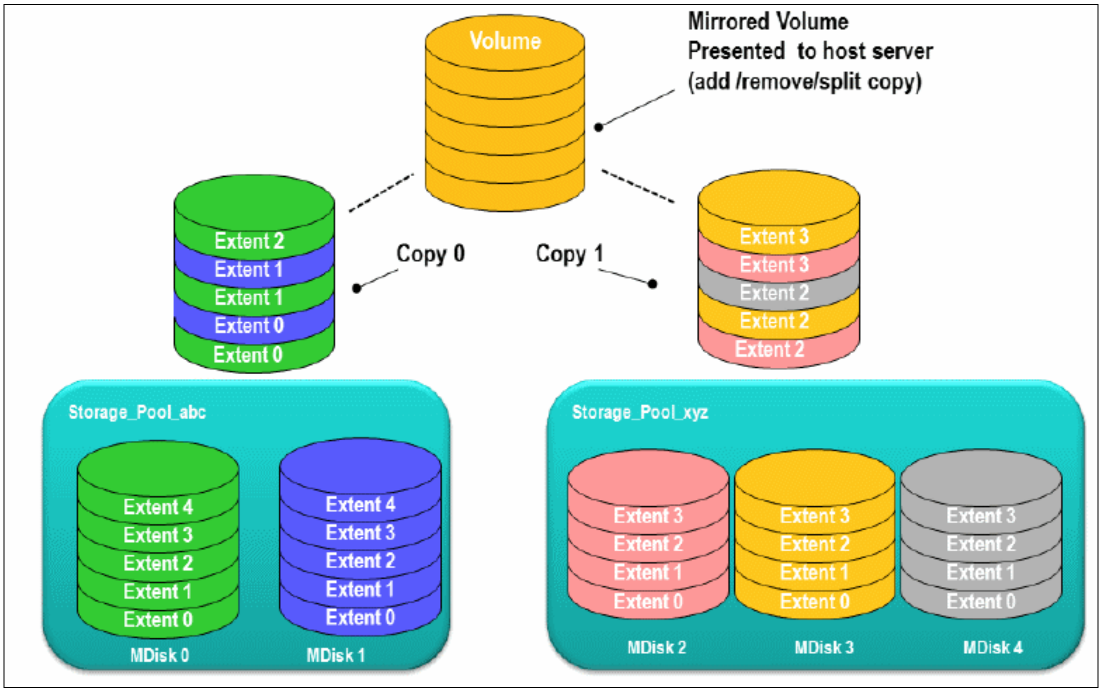


Figure 7-2 Volume Mirroring overview

Note the following points:

- 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 the Primary volume copy is available and synchronized, any reads from the volume are directed to it.
- If one of the mirrored volume copies is temporarily unavailable, for example, because the storage system that provides the storage pool is unavailable, the volume remains accessible to servers.
- The system remembers which areas of the volume are written and resynchronizes these areas when both copies are available.
- The secondary can service read I/O when the primary is offline without user intervention.

You can convert a non-mirrored volume into a mirrored volume by adding a copy. When a copy is added using this method, the cluster system synchronizes the new copy so that it is the same as the existing volume. 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.

> **Server access:** Servers can access the volume during the synchronization processes described.

You can use mirrored volumes to provide extra protection for your environment or to perform a migration. This offers several options:

- Export to Image mode

  This option allows you to move storage from *managed mode* to *image mode*. This is useful if you are using the SAN Volume Controller or Storwize V7000 as a migration device.

For example, suppose vendor A's product cannot communicate with vendor B's product, but you need to migrate existing data from vendor A to vendor B. Using "Export to image mode" allows you to migrate data using Copy Services functions and then return control to the native array, while maintaining access to the hosts.

- Import to Image mode

  This option allows you to import an existing storage MDisk or logical unit number (LUN) with its existing data from an external storage system, without putting metadata on it, so that the existing data remains intact. After you have imported it, all copy services functions can be used to migrate the storage to the other locations, while the data remains accessible to your hosts.

- Volume migration using Volume Mirroring and then using the Split into New Volume option

  This option allows you to use the available RAID 1 functionality. You create two copies of data that initially have a set relationship (one primary and one secondary). You then break the relationship (both primary and no relationship) to make them independent copies of data.

  You can use this option to migrate data between storage pools and devices. You might use this option if you want to move volumes to multiple storage pools.

  Mirroring consideration: You only can mirror one volume at a time.

- Volume migration using the Move to Another Pool option

  This option allows any volume to be moved between storage pools without any interruption to the host access. This option is effectively a quicker version of the Volume Mirroring and Split into New Volume option. You might use this option if you want to move volumes in a single step, or you do not have a volume mirror copy already.

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.

Quorum disk consideration: Mirrored volumes can be taken offline if there is no quorum disk available. This behavior occurs because synchronization status for mirrored volumes is recorded on the quorum disk. To protect against mirrored volumes being taken offline, follow the guidelines for setting up quorum disks.

7.1.1 Mirrored volume components

Note the following points regarding mirrored volume components:

- A mirrored volume is always composed of two copies (copy0 and copy1).
- A volume that is not mirrored consists of a single copy (which for reference could be copy 0 or copy 1).

A mirrored volume looks the same to upper-layer clients as a non-mirrored volume. That is, upper layers within the cluster software, such as FlashCopy and Metro Mirror/Global Mirror, and storage clients, do not know whether a volume is mirrored. They all continue to handle the volume as they did before without being aware of whether the volume is mirrored.

Figure 7-3 shows the attributes of a volume and Volume Mirroring.

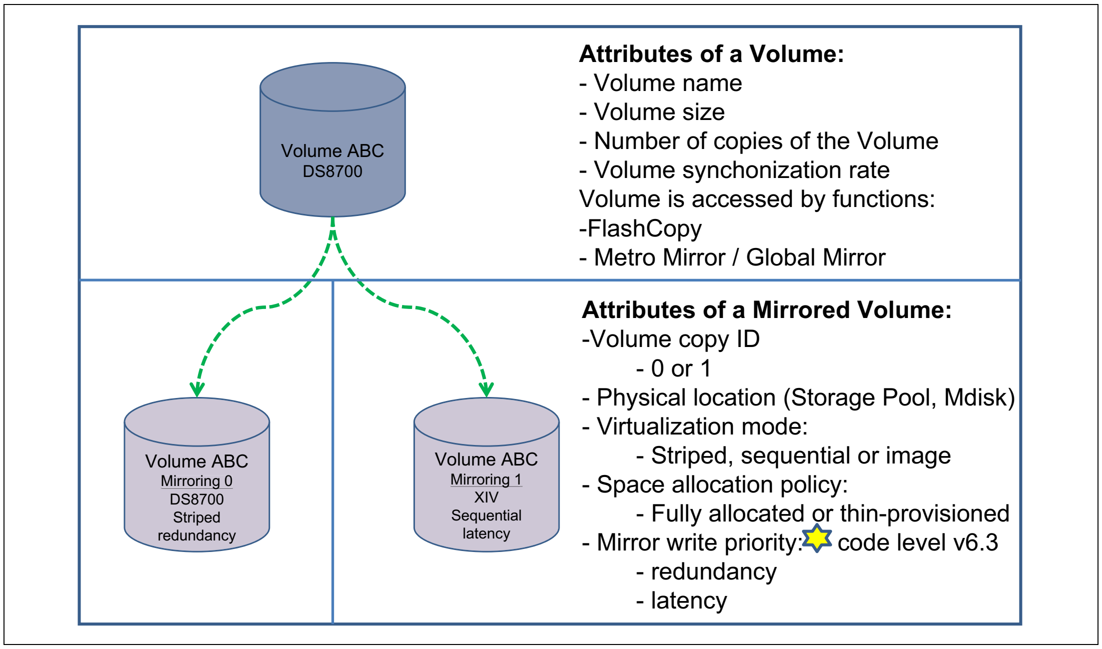


Figure 7-3 Attributes of a volume and Volume Mirroring

In Figure 7-3, XIV and DS8700 illustrate that a mirrored volume can utilize different storage devices.

Starting with SVC version 6.3.0 and the mirror write priority option, the Volume Mirroring feature has been enhanced with a tunable latency tolerance. This tolerance is designed to provide an option to give preference to losing the redundancy between the two copies. This tunable time-out value is either *latency* or *redundancy*. For example, when we have two copies of a volume and each of them is located in a different storage pool, we can specify a delay to have both copies fully consistent and by using the following attributes you can increase the time to wait for completion.

- Choosing latency means a copy that is slow to respond to a write I/O will become unsynchronized and the write I/O is completed if the other copy successfully writes the data.
- Choosing redundancy means a copy that is slow to respond to a write I/O causes completion of the write I/O to wait for the completion of the slow I/O, to maintain synchronization.

7.1.2 Volume Mirroring usage cases and characteristics

This section provides descriptions of common Volume Mirroring usage cases and the characteristics of each of them.

- Creating a mirrored volume.
  - The maximum number of copies is two.
  - Both copies will be created with the same virtualization policy.

    To have a volume mirrored using different policies, you need to add a volume copy with a different policy to a volume that has only one copy.

- Both copies can be located in different Storage Pools. The first Storage Pool specified will contain the primary copy.
- It is not possible to create a volume with two copies when specifying a set of MDisks.

- Add a volume copy to an existing volume.
  - The volume copy to be added can have a different space allocation policy.
  - Two existing volumes with one copy each cannot be merged into a single mirrored volume with two copies.
- Remove a volume copy from a mirrored volume.
  - The volume remains with only one copy.
  - It is not possible to remove the last copy from a volume.
- Split a volume copy from a mirrored volume and create a new volume with the split copy.
  - This function is only allowed when the volume copies are synchronized; otherwise, use the **`-force`** command.
  - It is not possible to recombine the two volumes again, after they have been split.
  - Adding and splitting in one workflow enables migrations that are not currently allowed.
  - The split volume copy can be used as a means for creating a point-in-time copy (clone).
- Repair/validate in three ways. This compares volume copies and:
  - Reports the first difference found.

    It can iterate by starting at a specific Logical Block Address (LBA) using the **`-startlba`** parameter.
  - Creates virtual medium errors where there are differences.
  - Corrects the differences found (reads from primary copy and writes to secondary copy).
- View to list volumes affected by a back-end disk subsystem being offline.
  - Assumes that a standard use is for mirror between disk subsystems.
  - Verifies that mirrored volumes will remain accessible if a disk system is being shut down.
  - Reports an error in case a quorum disk is on the back-end disk subsystem.
- Expand or shrink a volume.
  - This function works on both of the volume copies at once.
  - All volume copies always have the same size.
  - All copies must be synchronized before expanding or shrinking them.
- Delete a volume.

  When a volume gets deleted, all copies get deleted.
- Migration commands apply to a specific volume copy.
- Out-of-sync bitmaps share the bitmap space with FlashCopy and Metro Mirror/Global Mirror.

  Creating, expanding, and changing I/O groups might fail if there is insufficient memory.
- GUI views now contain volume copy identifiers.

7.1.3 Performance considerations of Volume Mirroring

Because the writes of mirrored volumes always occur to both copies, Mirroring Volumes puts more workload on the cluster, the back-end disk subsystems, and the connectivity infrastructure.

The mirroring is symmetrical, and writes are only acknowledged when the write to the last copy completes. The result is that if the volumes copies are located on Storage Pools with different performance characteristics, the slowest Storage Pool decides upon the performance of writes to the volume, which applies when writes have to be destaged to disk.

Recommendation: Locate volume copies of one volume on Storage Pools of the same or similar characteristics. Usually, if only good read performance is required, you can place the primary copy of a volume in a Storage Pool with better performance. Because the data is always only read from one volume copy, reads will not be faster than without Volume Mirroring.

Be aware, however, that this is only true when both copies are synchronized. If the primary is out of sync, then reads will be submitted to the other copy.

Synchronization between volume copies has a similar impact on the cluster and the back-end disk subsystems as FlashCopy or data migration. The synchronization rate is a property of a volume that is expressed as a value between 0 and 100. A value of 0 disables synchronization.

Table 7-1 shows the relationship between the *rate value* and the *data copied per second*.

Table 7-1 Relationship between the rate value and the data copied per second

User-specified rate attribute value per volume	Data copied/sec
0	Synchronization is disabled
1 - 10	128 KB
11 - 20	256 KB
21 - 30	512 KB
31 - 40	1 MB
41 - 50	2 MB **** 50% is the default value**
51 - 60	4 MB
61 - 70	8 MB
71 - 80	16 MB
81 - 90	32 MB
91 - 100	64 MB

Rate attribute value: The rate attribute is configured on each volume you want to mirror. The default value of a new volume mirror is 50%.

Mirrored Volume and I/O Time-out Configuration

The source volume has pointers to two copies (mirrored volume copies) of data each in different storage pools, and each write completes on both copies before the host receives I/O completion status.

For a synchronized mirrored volume, if a write I/O to a copy has failed or a long time-out has expired, then SVC has completed all available controller level ERPs. In this case, that copy is taken offline and goes out of sync. The volume remains online and continues to service I/O requests from the remaining copy.

Prior to v6.3.x, the time-out was brief to prevent hosts seeing extended I/O latency if one copy had issues. In v6.3.x, the time-out configuration can be set for *each* mirrored volume using the command **`chvdisk`**.

There are two possible values for I/O Time-out Configuration (attribute mirror_write_priority):

- Latency (default value): short time-out prioritizing low host latency. This option indicates a copy that is slow to respond to a write I/O goes out of sync if the other copy successfully writes the data.
- Redundancy: long time-out prioritizing redundancy. This option indicates a copy that is slow to respond to a write I/O may use the full Error Recovery Procedure (ERP) time. The response to the I/O is delayed until it completes to keep the copy in sync if possible.

Volume Mirroring ceases to use the slow copy for a period of between 4 to 6 minutes, and subsequent I/O data is not affected by a slow copy. Synchronization is suspended during this period. After the copy suspension completes, Volume Mirroring resumes, which allows I/O data and synchronization operations to the slow copy that will, typically, shortly complete the synchronization.

7.1.4 Bitmap space for out-of-sync volume copies

The grain size for the synchronization of volume copies is 256 KB. One grain takes up one bit of bitmap space. 20 MB of bitmap space supports 40 TB of mirrored volumes. This relationship is the same as the relationship for copy services (Global and Metro Mirror) and standard FlashCopy with a grain size of 256 KB.

Table 7-2 Relationship of bitmap space to Volume Mirroring address space

Function	Grain size in KB	1 byte of bitmap space gives a total of	4 KB of bitmap space gives a total of	1 MB of bitmap space gives a total of	20 MB of bitmap space gives a total of	512 MB of bitmap space gives a total of
Volume Mirroring	256	2 MB of volume capacity	8 GB of volume capacity	2 TB of volume capacity	40 TB of volume capacity	1024 TB of volume capacity

Shared bitmap space: This bitmap space on one I/O group is shared between Metro Mirror, Global Mirror, FlashCopy, and Volume Mirroring.

7.1.5 Synchronization status of volume copies

As soon as a newly volume is created with two copies, the state of copies are in *out-of-synchronization*.

The primary volume copy (located in the first specified Storage Pool) is defined as in sync and the secondary volume copy as out of sync.

The secondary copy will be synchronized through the synchronization process, which runs at the default synchronization rate of 50 (see more details in Table 7-1 on page 281), or at the defined rate while creating or modifying the volume.

There is a parameter **-format** that ensures both copies are overwritten with zeros, and after this the volume comes online and they can be considered as *synchronized copies*. Both copies are filled with zeros, so they are exactly the same.

You can specify that a volume is synchronized (**-createsync** parameter), even if it is not. Using this parameter can cause data corruption if the primary copy fails and leaves an unsynchronized secondary copy to provide data. Using this parameter can cause loss of read stability in unwritten areas if the primary copy fails, data is read from the primary copy, and then different data is read from the secondary copy. To avoid data loss or read stability loss, use this parameter only for a primary copy that has been formatted and not written to, and with the **-fmtdisk** parameter.

Another example usecase for **-createsync** is for a newly created mirrored volume where both copies are thin provisioned or compressed because no data has been written to disk and unwritten areas return zeroes (0).

If the synchronization between the volume copies has been lost, the resynchronization process is incremental. This means that only grains that have been written to need to be copied and then, get synchronized volume copies again.

The progress of the volume mirror synchronization can be obtained from the GUI or by using the **lsvdisksyncprogress** command.

7.2 Volume Mirroring infrastructure design examples

Volume Mirroring is a key function for Infrastructure Simplification, because it can enable the consolidation of Logical Volume Managers (LVMs) of storage clients into a function of the main storage subsystem, the SAN Volume Controller or Storwize V7000.

You can use Volume Mirroring to design a highly available (HA) storage infrastructure. It is not a function to ensure disaster tolerance on its own. Volume Mirroring can be used for Business Continuity.

7.2.1 HA design using a single SVC or Storwize V7000 cluster

Requirements of a design to be HA involve redundancy of the infrastructure to protect against failures of specific parts of it. The goal is that in these cases the service stays online (available), or the service can resume quickly if interrupted (without interrupting client functions).

For example, the SVC or Storwize V7000 is an HA storage controller that is able to provide its service to storage clients even in the event where one node fails to provide service. However, in a case of a node failure, this cluster is considered as "degraded." Every other failure that results in the remaining node failing to provide its service will result in the loss of access to that service.

Usually, back-end disk subsystems are themselves highly available designs. Nonetheless, sometimes these systems, or parts of them (RAID), fail to provide their service. Volume Mirroring addresses this issue. Also, you can use Volume Mirroring to ensure availability when performing service actions, such as a code update on back-end disk subsystems that do not support concurrent code updates.

With Volume Mirroring, the cluster writes the same data to two regions of storage that can reside in different storage pools from different back-end storage subsystems. This capability enables us to use the cluster in a manner which ensures the availability of its service even in the event that a back-end storage subsystem fails completely.

Figure 7-4 illustrates a simple environment using a single cluster with Volume Mirroring and two back-end disk subsystems.

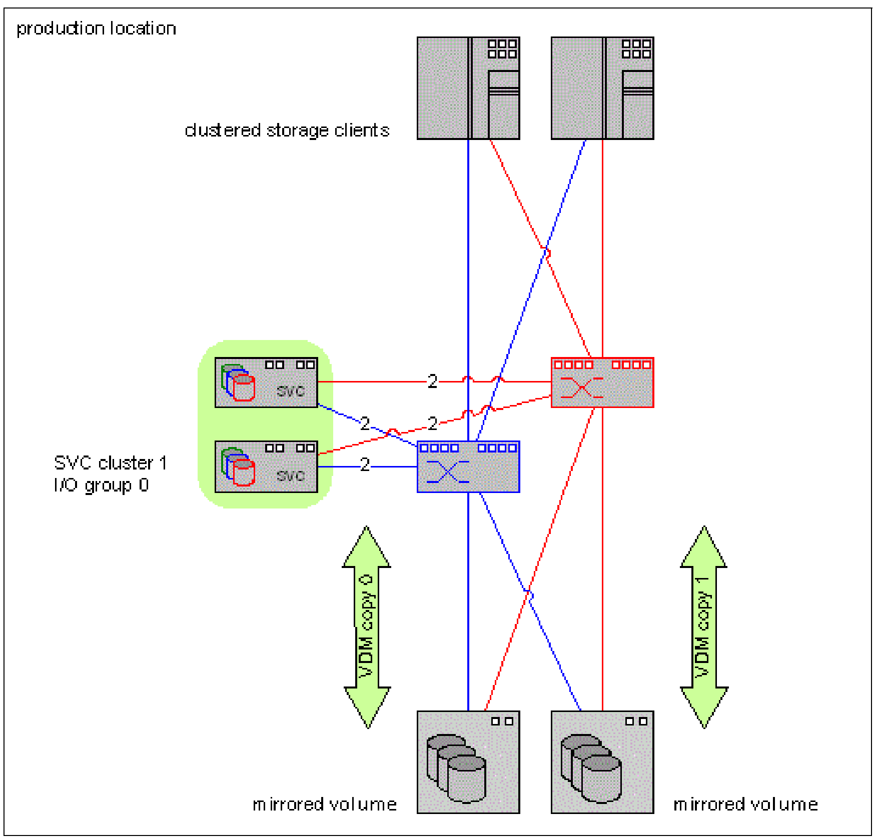


Figure 7-4 Single Cluster using Volume Mirroring on two back-end disk subsystems

7.2.2 HA and disaster-tolerant design using SVC or Storwize V7000 cluster

A *disaster* is usually defined as failure of a large part of the IT infrastructure caused by an event that interrupts IT service in an entire location. In the case of a disaster, if the redundancy is only deployed *within* this specific location, the redundant infrastructure cannot ensure the availability of the service. Also, service cannot be resumed quickly and, more importantly, with the most recent production data. A requirement for disaster tolerance is usually that at least one location is available from where the interrupted service can resume with the most recent production data. There are multiple levels of disaster tolerance. For instance, for certain services a longer downtime is acceptable than for other services.

If a disaster is declared, the location that is not affected must be able to restart the service of the affected location. We will call these two locations the “production location” and the “disaster location.” In the disaster location, we have the same infrastructure installed and ready to provide service that is installed on the production location. Sometimes the disaster location’s capability is less powerful, but it is designed with the same HA requirements as the production location.

To design a disaster-tolerant infrastructure, use one cluster at the production location and another cluster at the disaster location. Use either Metro Mirror or Global Mirror solutions between these Clusters. This way, you have a production environment and the same data in the disaster location ready to be used. The second cluster in the disaster location is not degraded if a disaster strikes in the production location, because it is an independent cluster.

The disaster tolerance enabling functions Metro Mirror/Global Mirror can be used together with the HA enabling function Volume Mirroring. This way, the cluster in the production location provides highly available storage, while at the same time the production data gets copied to the second cluster in the disaster location. If needed, the cluster at the disaster location can also use Volume Mirroring.

Figure 7-5 illustrates an HA and disaster-tolerant design using two clusters with Volume Mirroring on the production location.

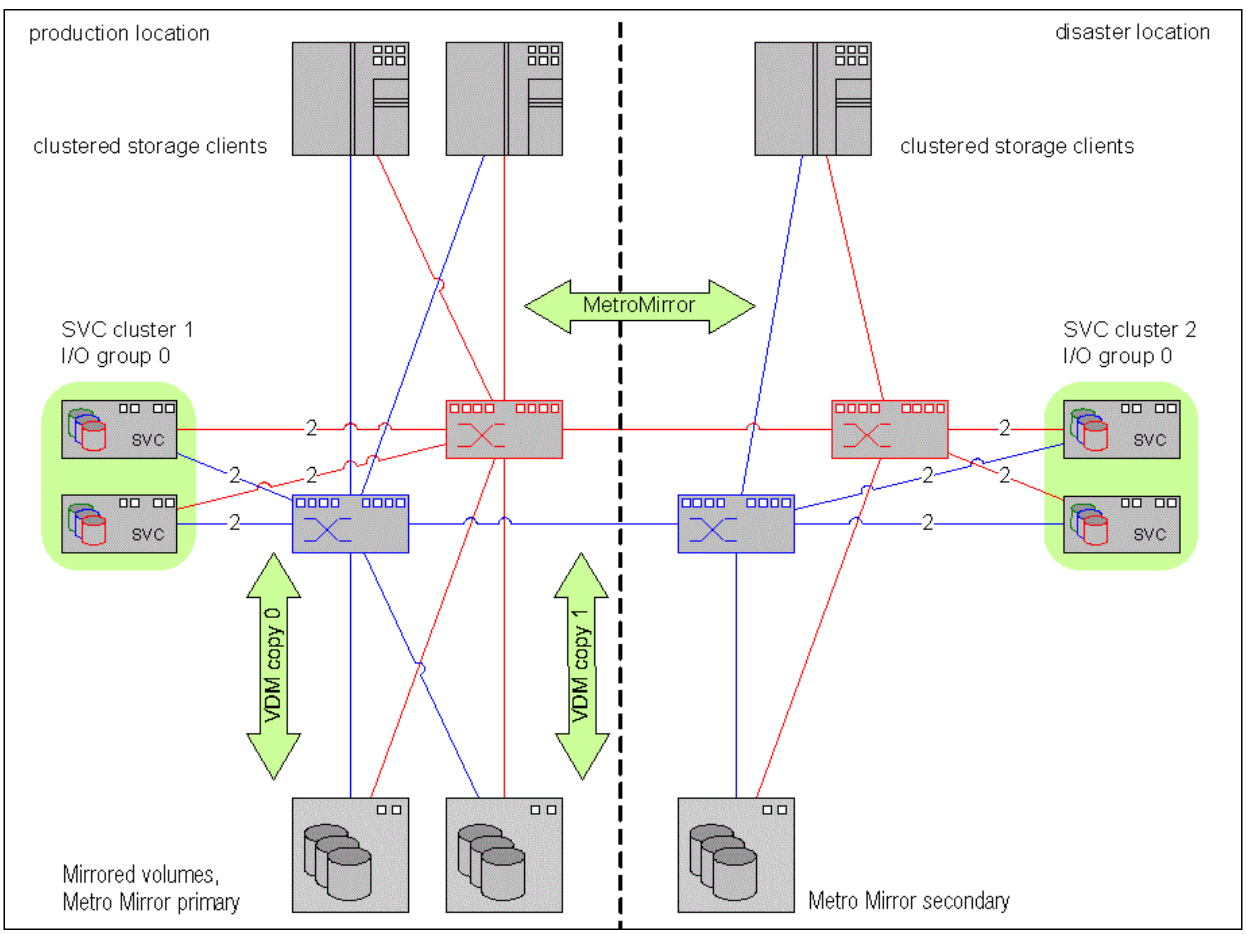


Figure 7-5 Two clusters using Metro Mirror: Production site is using Volume Mirroring on two back-end disk systems using mirrored copies

7.2.3 Quorum disks

Although quorum disks have little to do with Volume Mirroring, Volume Mirroring makes use of the metadata store and it is linked into High Availability system configuration planning so we discuss it here.

The system uses the quorum disk for two purposes:

- As a tiebreaker in the event of a SAN fault (when exactly half of the nodes that were previously members of the system are present)
- To hold a copy of important system configuration data.

Slightly over 256 MB is reserved for this purpose on each quorum disk candidate. Only one active quorum disk exists in a system. However, the system uses three MDisks as quorum disk candidates. The system automatically selects the actual active quorum disk from the pool of assigned quorum disk candidates.

If a tiebreaker condition occurs, the one-half portion of the system nodes that is able to reserve the quorum disk after the split has occurred locks the disk and continues to operate. The other half stops its operation. This design prevents both sides from becoming inconsistent with each other.

When MDisks are added to the SVC or Storwize systems, the system checks the MDisk to see if it can be used as a quorum disk. If the MDisk fulfills the requirements, the cluster will assign the three first MDisks that are added to the system as quorum candidates. One of these MDisks is selected as the active quorum disk.

7.2.4 Quorum disk considerations

Note the following points regarding quorum disks.

- Quorum disk requirements:

  To be considered eligible as a quorum disk, an LUN must meet the following criteria:

  - It must be presented by a disk subsystem that is supported to provide cluster quorum disks.
  - It has been manually allowed to be a quorum disk candidate using the **`svctask chcontroller -allow_quorum yes`** command.
  - It must be in managed mode (no image mode disks).
  - It must have sufficient free extents to hold the system state information, plus the stored configuration metadata.
  - It must be visible to all of the nodes in the system.
- Quorum disk placement:

  If possible, the cluster will place the quorum candidates on separate disk subsystems. After the quorum disk has been selected, however, no attempt is made to ensure that the other quorum candidates are presented through separate disk subsystems.

  > **Important**: Quorum disk placement verification and adjustment to separate storage systems, if possible, reduce the dependency from a single storage system and can increase quorum disk availability significantly.

You can list the quorum disk candidates and the active quorum disk in a system by using the **`svcinfo lsquorum`** command.

When the set of quorum disk candidates has been chosen, it is fixed. However, a new quorum disk candidate can be chosen in one of these conditions:

- When the administrator requests that a specific MDisk is to become a quorum disk by using the **`svctask chquorum`** command
- When an MDisk that is a quorum disk is deleted from a storage pool
- When an MDisk that is a quorum disk changes into an image mode

An offline MDisk will not be replaced as a quorum disk candidate.

For disaster recovery purposes, a system needs to be regarded as a single entity, so the system and the quorum disk need to be collocated.

Since SVC version 6.2, the quorum disk is automatically selected by the system with the following main objectives:

- Maintain the same common code base for the SVC/Storwize family.
- Make clustering transparent for Storwize V7000 customers.
- Monitor the cluster continuously and select the quorum option for best availability.

There are special considerations concerning the placement of the active quorum disk for stretched cluster configurations. The details are available at the following address:

`http://www-01.ibm.com/support/docview.wss?rs=591&uid=ssg1S1003311`

Importance of quorum disks: Running a cluster system without a quorum disk can seriously affect your operation. A lack of available quorum disks for storing metadata will prevent any migration operation (including a forced MDisk delete).

Mirrored volumes can be taken offline if there is no quorum disk available. This behavior occurs because the synchronization status for mirrored volumes is recorded on the quorum disk.

During the normal operation of the system, the nodes communicate with each other. If a node is idle for a few seconds, a heartbeat signal is sent to ensure connectivity with the system. If a node fails for any reason, the workload that is intended for it is taken over by another node until the failed node has been restarted and readmitted into the system (which happens automatically). If the microcode on a node becomes corrupted, resulting in a failure, the workload is transferred to another node. The code on the failed node is repaired, and the node is readmitted into the system (again, all automatically).

7.2.5 High availability design using an SVC Stretched Cluster configuration

You can split an SVC cluster so that one node of each I/O group is installed in one of two locations. This is referred to as a Stretched Cluster configuration. This design enables high availability across two sub-locations.

Volume mirroring provides a consistent data copy in both sites. If one storage subsystem fails, the remaining subsystem processes the I/O requests. The combination of SVC node distribution in two independent data centers and a copy of data in two independent data centers creates a new level of availability.

All SVC nodes and the storage system in a single site might fail; the other SVC nodes take over the server load using the remaining storage subsystems. The volume ID, behavior, and assignment to the server are still the same. No server reboot, no failover scripts, and thus no script maintenance are required.

> **Storwize 7000 and Stretched Cluster configuration:** Currently the Storwize V7000 system does not support Stretched Cluster configuration.

Because it is still a single cluster, this design alone does not meet the standards of disaster tolerance. Use Metro Mirror/Global Mirror to create a system that can tolerate a catastrophic event. If a disastrous event in one location occurs, the SVC will enter a degraded state (only one node per I/O group left, cache write through, and no further redundancy).

> **Automatic quorum disk selection consideration:** If using an SVC Stretched Cluster configuration, the automatic quorum disk selection (starting with version 6.2) might result in a quorum disk configuration that violates the recommendations for SVC Stretched Cluster design. Disable it for each quorum disk by using the command **chquorum** as shown:
>
> ```
> chquorum -override yes quorum_id
> ```

Achieving high availability across sub-locations

Using a Stretched Cluster setup, you can achieve a higher level of availability in case of events that cause multiple failures of devices within one location but do not meet your criteria for declaring a full disaster. Within your production location you can set up two fault domains, for instance, fire compartments, and distribute your infrastructure across these fault domains in a way that the devices installed in only one location suffice to keep up the service.

This design means half of the back-end disk subsystems, one of the two redundant fabrics of the SAN, one node of each I/O group, and one node of clustered storage clients exist in each sub-location. This way, even if the majority of the equipment in one of the fault domains or even the whole fire compartment fails to provide service, this design increases the probability that your service will not be disrupted.

FCIP and WDM

You can use distance extension through FCIP or WDM for quorum site connectivity with the requirements as described here. In any case, the connections must be reliable. It is strictly required that the links from both production sites to the quorum site are independent and do not share any long-distance equipment.

FCIP routers can be used for quorum disk connections with SVC version 6.3.0 or later with the following requirements:

- The FCIP router device is supported for SVC remote mirroring (Metro Mirror or Global Mirror).
- The maximal round trip delay does not exceed 80 ms, that means 40 ms each direction.
- A minimal bandwidth of 2 MegaBytes per second (MBps) is guaranteed for node-to-quorum traffic.

  > **FCIP links configuration:** Configure FCIP links so that they do not carry ISLs, to avoid fabric topology changes in case of IP errors.

- No inter-switch link ISL configuration:
  – Passive Wavelength Division Multiplexing (WDM) devices can be used between both sites.

– No ISLs can be used between SVC nodes (this is similar to the SVC 5.1 supported configuration).
– The distance extension is to up to 40 km (24.8 miles).

- ISL configuration:
  – ISLs are allowed between SVC nodes (not allowed with releases earlier that 6.3).
  – The maximum distance is similar to Metro Mirror (MM) distances.
  – The physical requirements are similar to MM requirements.
  – ISL distance extension is allowed with active and passive WDM devices.

This chapter explores the characteristics of both configurations.

7.2.6 Non-ISL configuration

In a non-ISL configuration, each SVC I/O Group consists of two independent SVC nodes. In contrast to a standard SVC environment, nodes from the same I/O Group are not placed close together. They are distributed across two sites. If a node fails, the other node in the same I/O Group takes over the workload, which is standard in an SVC environment.

However, you must consider that a Stretched Cluster typically requires a special setup and performs differently. In a Stretched Cluster environment, the SVC nodes from the same I/O Group reside in two sites. A third quorum location is required for handling “split brain” scenarios.

Figure 7-6 shows an example of a non-ISL Stretched Cluster configuration as it is currently supported in SVC V6.3.

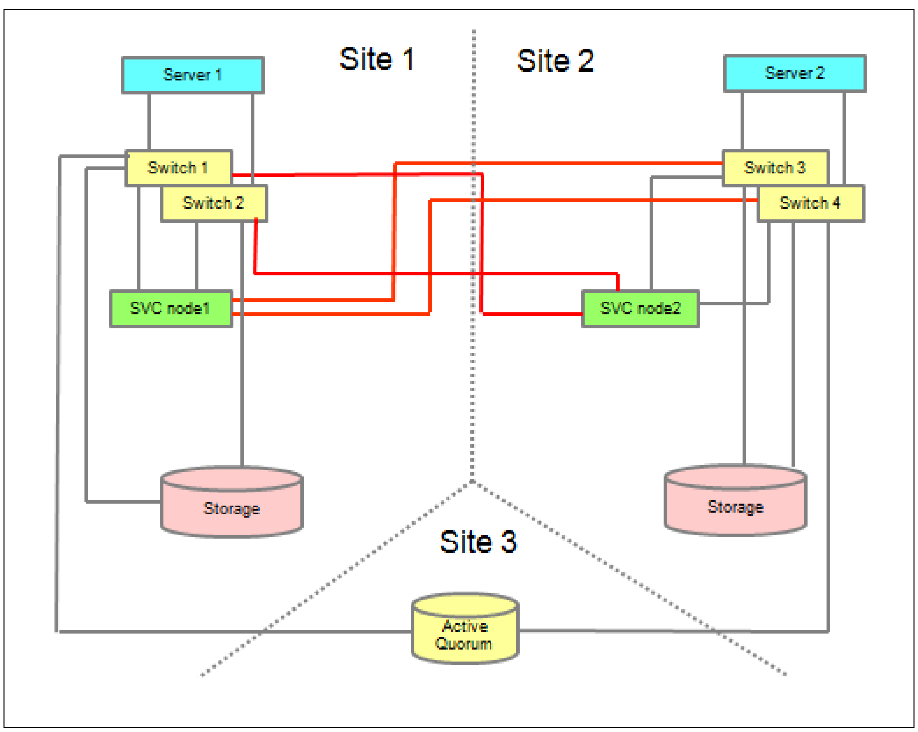


Figure 7-6 Standard SVC 6.2 environment using Volume Mirroring

The Stretched Cluster uses the SVC Volume Mirroring functionality. Volume Mirroring allows the creation of one volume with two copies of MDisk extents; there are not two volumes with the same data. The two data copies can be in different storage pools. Thus, Volume Mirroring can minimize the effect on volume availability if one set of MDisks goes offline. The resynchronization between both copies after recovering from a failure is incremental; SVC starts the resynchronization process automatically.

Like a standard volume, each mirrored volume is owned by one I/O Group with a preferred node. Thus, the mirrored volume goes offline if the whole I/O Group goes offline. The preferred node performs all I/O operations, which means reads and writes. The preferred node can be set manually.

The quorum disk keeps the status of the mirrored volume. The last status (in sync or not in sync) and the definition of the primary and secondary volume copy are saved there. Thus, an active quorum disk is required for Volume Mirroring. To ensure data consistency, the cluster disables all mirrored volumes if there is no access to any quorum disk candidate any longer. Therefore, quorum disk placement is an important area with Volume Mirroring and Stretched Cluster configuration.

Preferred practices:

- Drive read I/O to the local storage system.
- For distances less than 10 km (6.2 miles), drive the read I/O to the faster of the two disk subsystems if they are not identical.
- Take long distance links into account.
- The preferred node must stay at the same site as the server accessing the volume.
- The Volume Mirroring primary copy must stay at the same site as the server accessing the volume to avoid any potential latency effect where the longer distance solution will be implemented.

In many cases, no independent third site is available. It is possible to use an already existing building or computer room from the two main sites to create a third, independent failure domain. There are several considerations:

- The third failure domain needs an independent power supply or uninterruptible power supply. If the hosting site fails, the third failure domain must continue to operate.
- A separate storage controller for the active SVC quorum disk is required. Otherwise, the cluster loses multiple quorum disk candidates at the same time if a single storage subsystem fails.
- Each site (failure domain) must be placed in a different location in case of fire.
- Fibre Channel (FC) cabling must not go through another site (failure domain). Otherwise, a fire in one failure domain might destroy the links (and break access) to the SVC quorum disk.

As shown in Figure 7-6 on page 289, the setup is similar to a standard SVC environment, but the nodes are distributed to two sites.

SVC nodes and data are equally distributed across two separate sites with independent power sources, which are named as separate failure domains (Failure Domain 1 and Failure Domain 2). The quorum disk is located in a third site with a separate power source (Failure Domain 3).

Each I/O Group requires four dedicated fiber optic links between site 1 and site 2.

If the non-ISL configuration is implemented over a 10 km (6.2 mile) distance, passive WDM devices (without power) can be used to pool multiple fiber optic links with different wavelengths in one or two connections between both sites. SFPs with different wavelengths ("colored SFPs", that is, small form-factor pluggable (SFPs) that are used in Coarse Wave Division Multiplexing (CWDM) devices are required here.

The maximum distance between both major sites is limited to 40 km (24.8 miles).

Because you have to prevent the risk of burst traffic (due to the lack of buffer-to-buffer credits), the link speed must be limited. The link speed depends on the cable length between the nodes in the same I/O Group, as shown in Table 7-3.

Table 7-3 SVC/Storwize code level lengths and speed

SVC code level	Minimum length	Maximum length	Maximum link speed
>= SVC 5.1	>= 0 km	= 10 km (6.2 miles)	8 Gbps
>= SVC 6.3	>= 10 km (6.2 miles)	= 20 km (12.4 miles)	4 Gbps
>= SVC 6.3	>= 20 km (12.4 miles)	= 40 km (24.8 miles)	2 Gbps

The quorum disk at the third site must be FC-attached. Fibre Channel over IP (FCIP) can be used if the round-trip delay time to the third site is always less than 80 ms, which is 40 ms in each direction.

Figure 7-7 on page 292 shows a detailed diagram where passive WDMs are used to extend the links between site 1 and site 2.

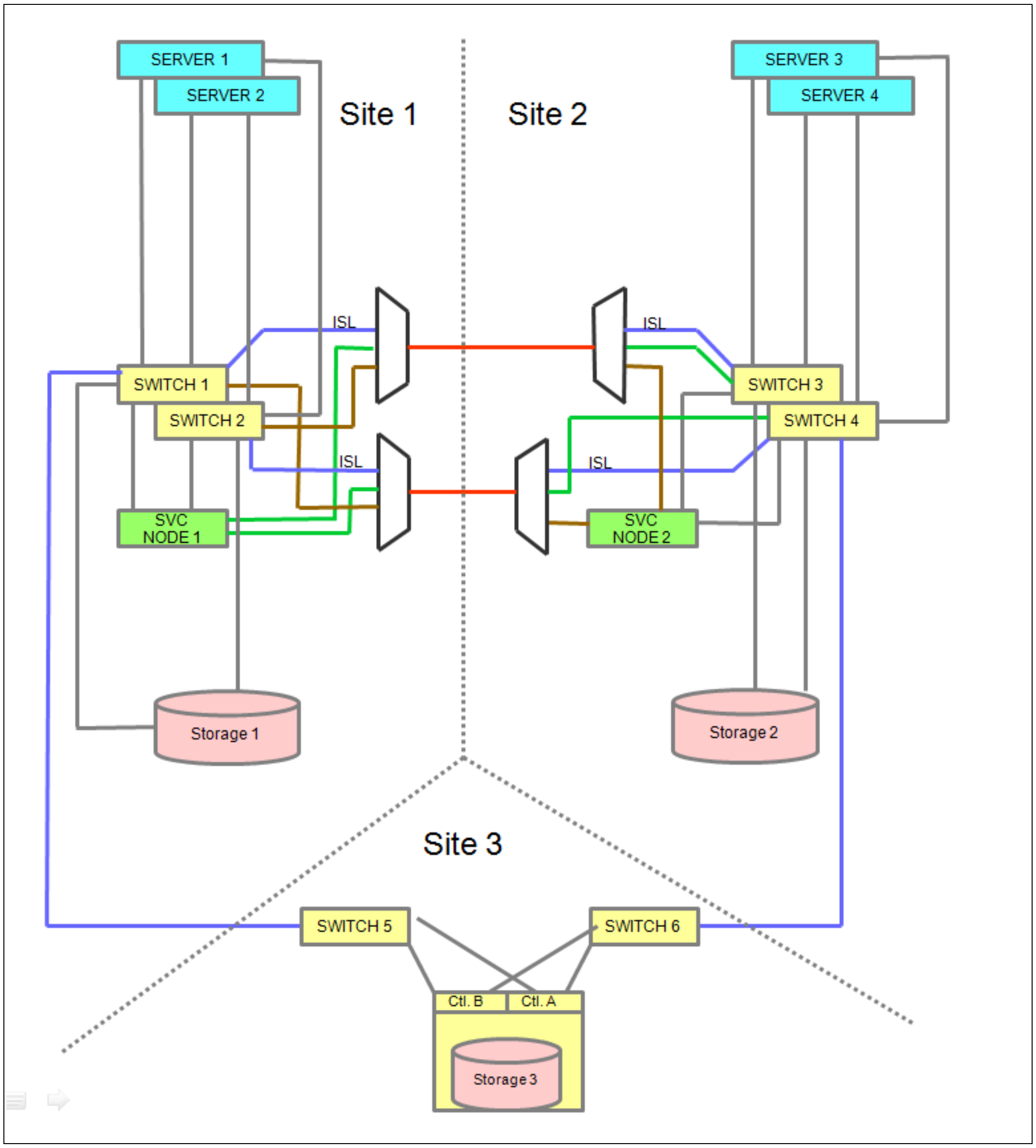


Figure 7-7 Connection with passive WDMs

The best performance server in site 1 must access the volumes in site 1 (preferred node and primary copy in site 1). SVC Volume Mirroring copies the data to storage 1 and storage 2. A similar setup must be implemented for the servers in site 2 with access to the SVC node in the site 2.

The configuration that is shown in Figure 7-7 covers several failover cases:

- Power off FC switch 1: FC switch 2 takes over the load and routes I/O to Cluster node 1 and Cluster node 2.
- Power off Cluster node 1: Cluster node 2 takes over the load and serves the volumes to the server. Cluster node 2 changes the cache mode to write-through to avoid data loss in case Cluster node 2 also fails.
- Power off storage 1: The Cluster waits a short time (15 - 30 seconds), pauses volume copies on storage 1, and continues I/O operations using the remaining volume copies on storage 2.

- Power off site 1: The server has no access to the local switches any longer, causing access loss. You optionally can avoid this access loss by using additional fiber optic links between site 1 and site 2 for server access.

The same scenarios are valid for site 2. And similar scenarios apply in a mixed failure environment, for example, failure of switch 1, Cluster node 2, and storage 2. No manual failover/failback activities are required, because the Cluster performs the failover and failback operation.

Benefits

- Business continuity solution distributed across two independent data centers.
- Configuration similar to a standard SVC clustered system.
- Limited hardware effort: Passive WDM devices can be used, but are not required.

Requirements

- Four independent fiber optic links for each I/O Group between both data centers is required.
- LW SFPs with support over long distance is required for direct connection to a remote storage area network (SAN).
- Optional use of passive WDM devices.
- Passive WDM device: No power is required for operation.
- "Colored SFPs" are required to make different wavelength available.
- "Colored SFPs" must be supported by the switch vendor.
- Using two independent fiber optic links between site 1 and site 2 is advisable.
- Third site for quorum disk placement is required.
- Quorum disk storage system must use FC for attachment with similar requirements, such as Metro Mirror storage (80 ms round-trip delay time, which is 40 ms in each direction)

Bandwidth reduction

Buffer credits, which are also called buffer-to-buffer (BB) credits, are used as a flow control method by FC technology. They represent the number of frames that a port can store.

Thus, buffer-to-buffer credits are necessary to have multiple FC frames in parallel in-flight. An appropriate number of buffer-to-buffer credits are required for optimal performance. The number of buffer credits to achieve the maximum performance over a given distance actually depends on the speed of the link:

- 1 buffer credit = 2 km (1.2 miles) at 1 Gigabit per second (Gbps)
- 1 buffer credit = 1 km (.62 miles) at 2 Gbps
- 1 buffer credit = 0.5 km (.3 miles) at 4 Gbps
- 1 buffer credit = 0.25 km (.15 miles) at 8 Gbps

The preceding guidelines give the minimum numbers. The performance drops if there are not enough buffer credits, according to the link distance and link speed (Table 7-4 on page 294).

Table 7-4 FC link speed buffer-to-buffer and distance

FC link speed	B2B credits for 10 km (6.2 miles)	Distance with eight credits
1 Gbit/s	5	16 km (9.9 miles)
2 Gbit/s	10	8 km (4.9 miles)
4 Gbit/s	20	4 km (2.4 miles)
8 Gbit/s	40	2 km (1.2 miles)

The number of buffer-to-buffer credits provided by an Cluster FC host bus adapter (HBA) is limited. An HBA of a 2145-CF8 node provides 41 buffer credits, which is sufficient for a 10 km (6.2 miles) distance at 8 Gbps. The Cluster adapters in all earlier models provide only eight buffer credits, which is enough only for a 4 km (2.4 miles) distance with a 4 Gbps link speed. These numbers are determined by the hardware of the HBA and cannot be changed.

It is advisable to use 2145-CF8 or CG8 nodes for distances longer than 4 km (2.4 miles) to provide enough buffer-to-buffer credits at a reasonable FC speed.

7.2.7 ISL configuration

Where a longer distance beyond 40 km (24.8 miles) between site 1 and site 2 is required, a new configuration must be applied. The setup is quite similar to a standard cluster environment, but the nodes are allowed to communicate over long distance using ISL links between both sites using active or passive WDM and a different SAN configuration.
Figure 7-8 on page 295 shows a detailed diagram that relates to a configuration with active or passive WDM.

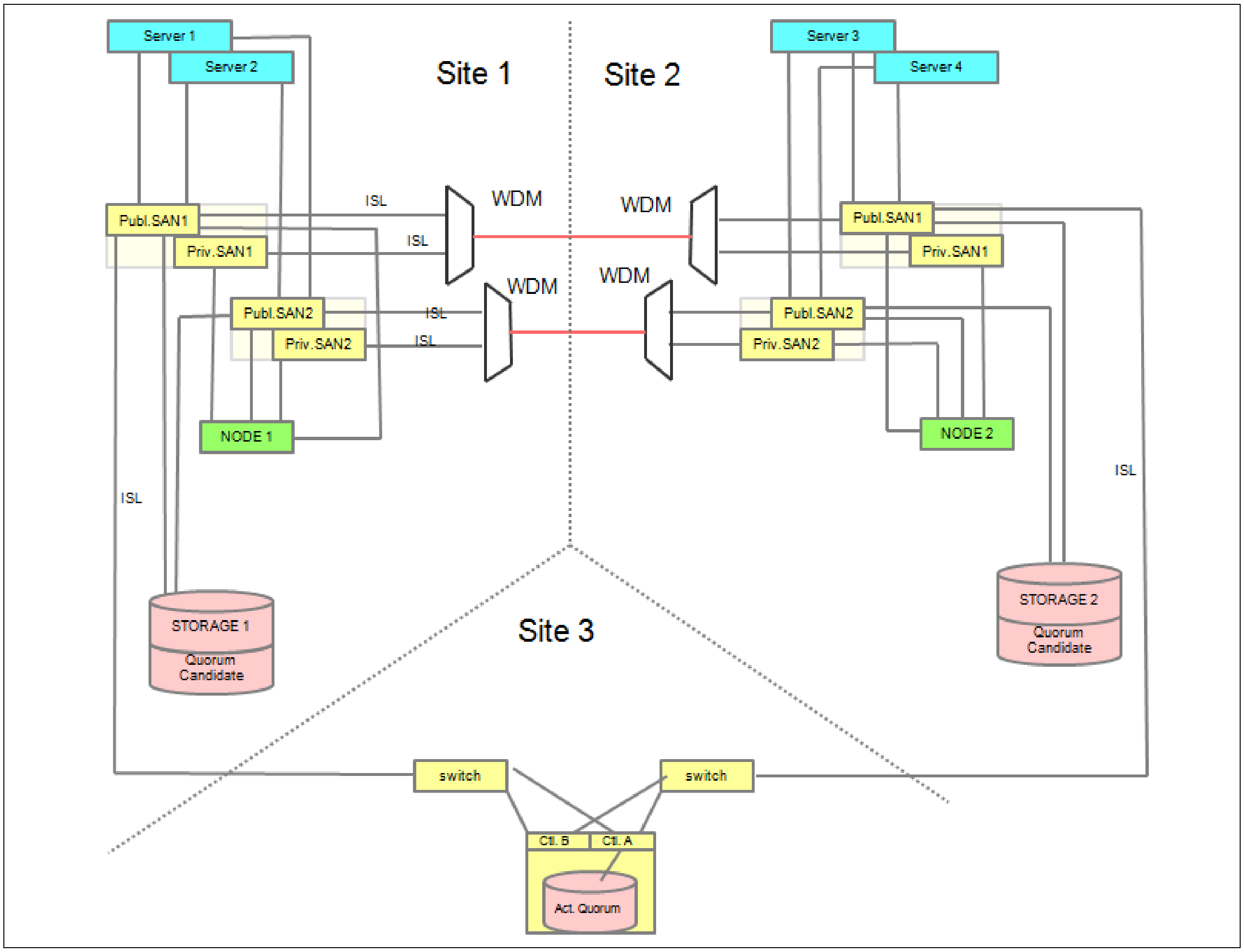


Figure 7-8 Connection with active/passive WDM and ISL

The Stretched Cluster configuration that is shown in Figure 7-8 supports distances of up to 300 km (186.4 miles), which is the same as the recommended distance for Metro Mirror.

Technically, the SVC will tolerate a round-trip delay of up to 80 ms between nodes. Cache mirroring traffic rather than Metro Mirror traffic is sent across the intersite link and data is mirrored to back-end storage using Volume Mirroring.

Data is written by the preferred node to both the local and remote storage. The SCSI write protocol results in two round-trips. This latency is hidden from the application by the written cache.

The Stretched Cluster is often used to move the workload between servers at separate sites. VMotion or the equivalent can be used to move applications between servers; therefore, applications no longer necessarily issue I/O requests to the local cluster nodes.

SCSI write commands from hosts to remote cluster nodes result in an additional two round-trips worth of latency that is visible to the application. For Stretched Cluster configurations in a long distance environment, it is advisable to use the local site for host I/O. Certain switches and distance extenders use extra buffers and proprietary protocols to eliminate one of the round-trip's worth of latency for SCSI write commands.

These devices are already supported for use with the cluster. They do not benefit or affect internode communication, but they benefit the host-to-remote SVC I/Os and SVC-to-remote storage controller I/Os.

Stretched Cluster with ISL configuration requirements

A Stretched Cluster with ISL configuration must meet the following requirements:

- Four independent, extended SAN fabrics are shown in Figure 7-8 on page 295. Those fabrics will be named Public SAN1, Public SAN2, Private SAN1, and Private SAN2. Each Public or Private SAN can be created with a dedicated FC switch or director. Or, they can be a virtual SAN in a CISCO or Brocade FC switch or director.
- Two ports per SVC/Storwize node attach to the private SANs.
- Two ports per SVC/Storwize node attach to the public SANs.
- SVC/Storwize Volume Mirroring exists between site 1 and site 2.
- Hosts and storage attach to the public SANs.
- The third site quorum attaches to the public SANs.

  Figure 7-9 shows the possible configurations with a virtual SAN.

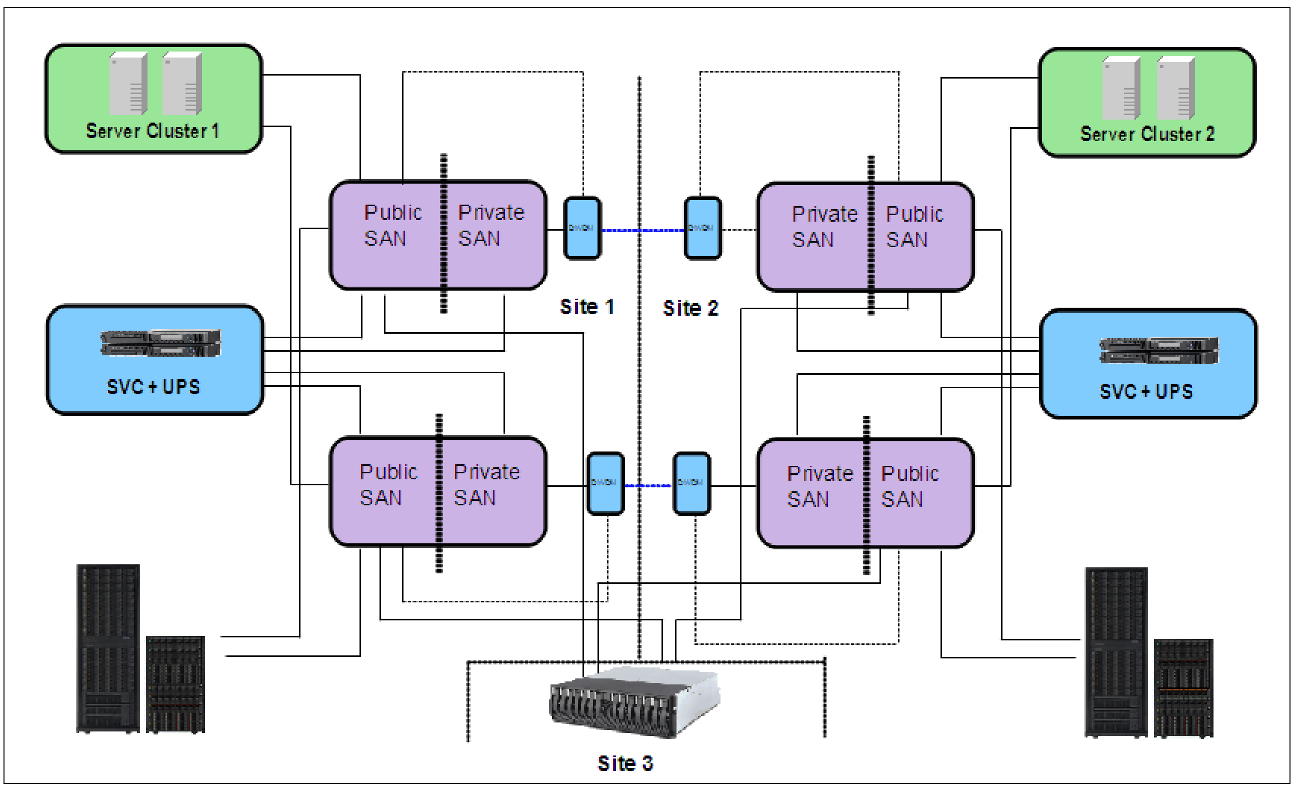


Figure 7-9 ISL configuration with a virtual SAN

Figure 7-10 shows the possible configurations with a physical SAN.

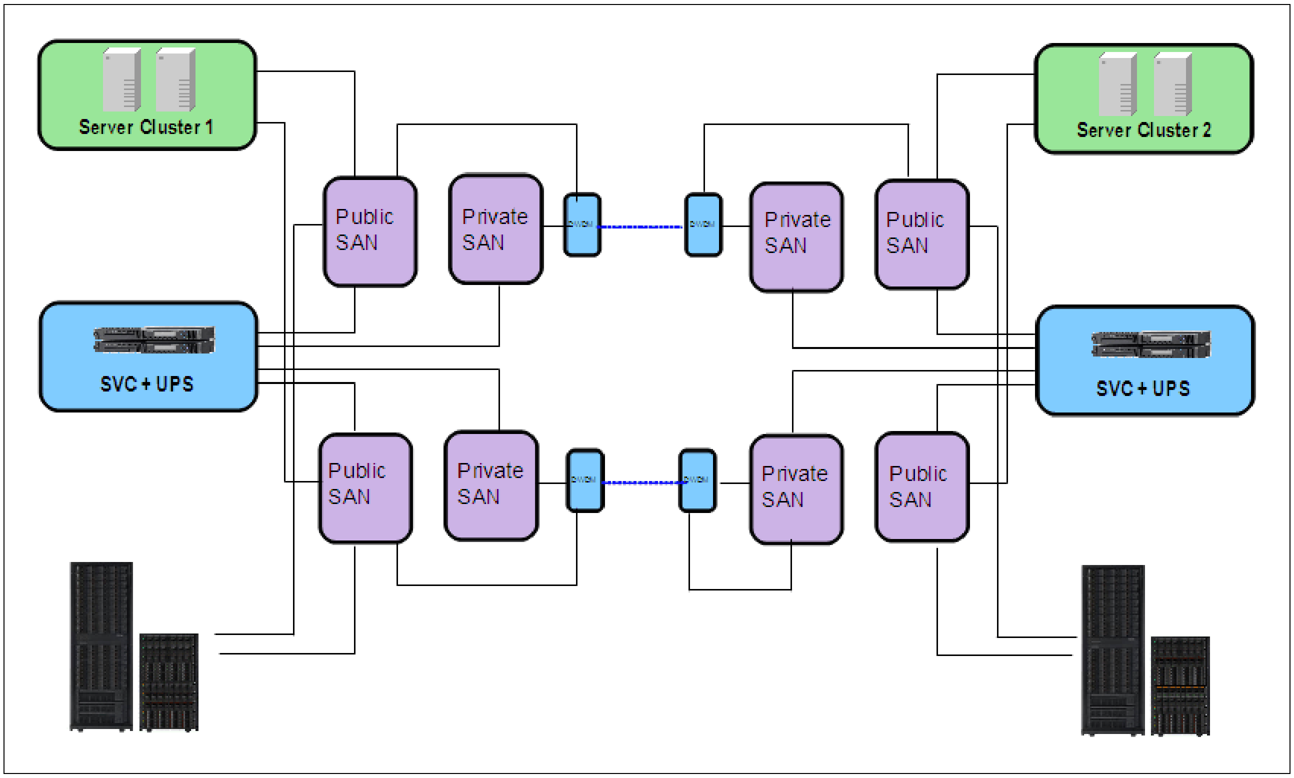


Figure 7-10 ISL configuration with a physical SAN

- Use a third site to house a quorum disk. Connections to the third site can be through FCIP because of the distance (no FCIP or FC switches were shown in the previous layouts for simplicity). In many cases, no independent third site is available.

  It is possible to use an already existing building from the two main sites to create a third, independent failure domain, but there are several considerations:

  – The third failure domain needs an independent power supply or uninterruptible power supply. If the hosting site failed, the third failure domain should continue to operate.

  – Each site (failure domain) needs to be placed in separate fire compartments.

  – FC cabling must not go through another site (failure domain). Otherwise, a fire in one failure domain destroys the links and breaks the access to the SVC quorum disk.

  Applying these considerations, the SVC clustered system can be protected, although two failure domains are in the same building.

  Consider an IBM Advanced Technical Support (ATS) review or processing a request for price quotation (RRQ)/Solution for Compliance in a Regulated Environment (SCORE) to review the proposed configuration.

  The storage system that provides the quorum disk at the third site must support extended quorum disks.

  Storage systems that provide extended quorum support are listed at the following website:

  http://www-01.ibm.com/support/docview.wss?uid=ssg1S1003907

- Four active/passive WDMs, two per each site, are needed to extend the public and private SAN over a distance.

- Place independent storage systems at the primary and secondary sites, and use Volume Mirroring to mirror the host data between storage systems at the two sites.
- The cluster nodes that are in the same I/O Group must be located in two remote sites.

For further detail about how configure the Stretched Cluster and Quorum disks, see the following IBM Redbooks publications:

- *Implementing the IBM System Storage SAN Volume Controller V6.3*, SG24-7933
- *Implementing the IBM Storwize V7000 V6.3*, SG24-7938

8

Implementing Volume Mirroring

This chapter explains how to use Volume Mirroring using the Graphical User Interface (GUI) and Command-line Interface (CLI). It is assumed you have a fully functional cluster (SAN Volume Controller or Storwize V7000) environment.

8.1 Managing bitmap space for Volume Mirroring using the GUI

Before working with tasks to manage Volume Mirroring, you need to configure bitmap space for Volume Mirroring.

> **Automatic bitmap allocations:** SVC 6.2.0 introduced automatic bitmap allocations for creating mirrored volumes using the GUI or CLI (**`mkvdisk`** or **`addvdiskcopy`**).

Each I/O group has an independent bitmap space configuration. The maximum value for all features (FlashCopy, Global and Metro Mirror, Volume Mirroring and RAID) is 552 MB.

To reserve bitmap space for Volume Mirroring, navigate in the GUI to **Monitoring** → **System** as shown in Figure 8-1.

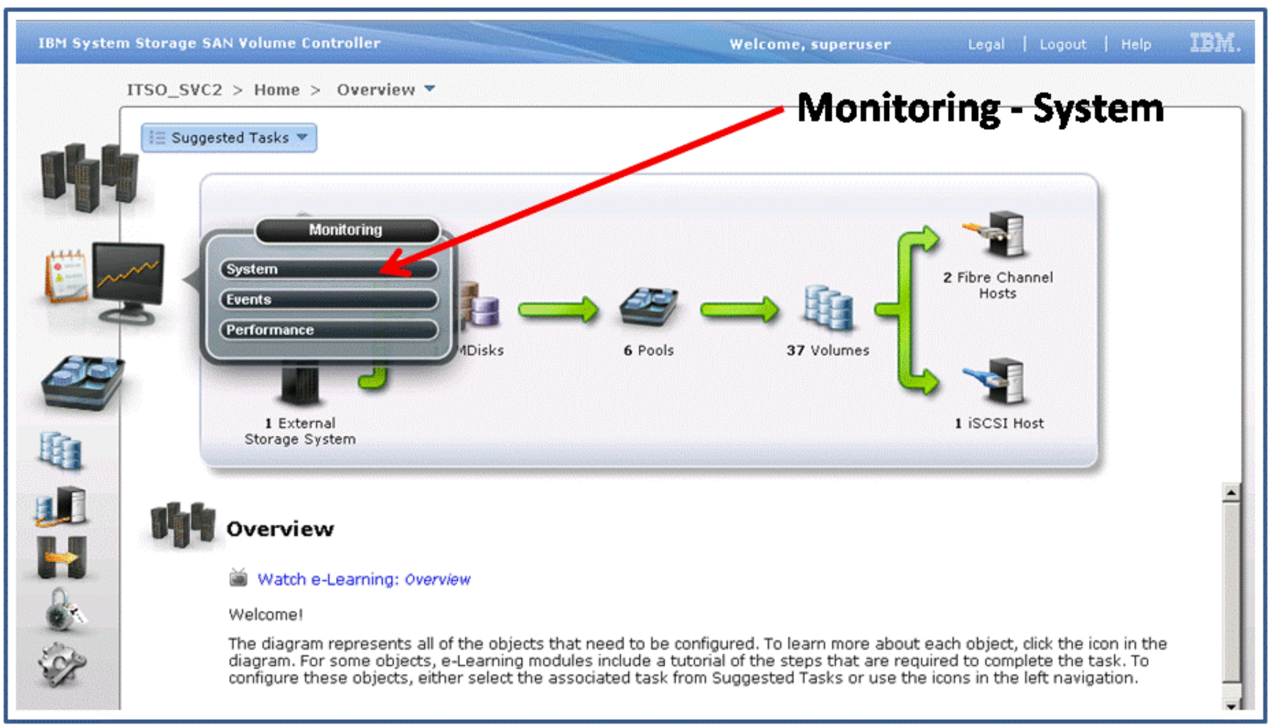


Figure 8-1 Access or change bitmap space configuration

Select the I/O group for which you need to view or change bitmap space properties, as shown in Figure 8-2 on page 301.

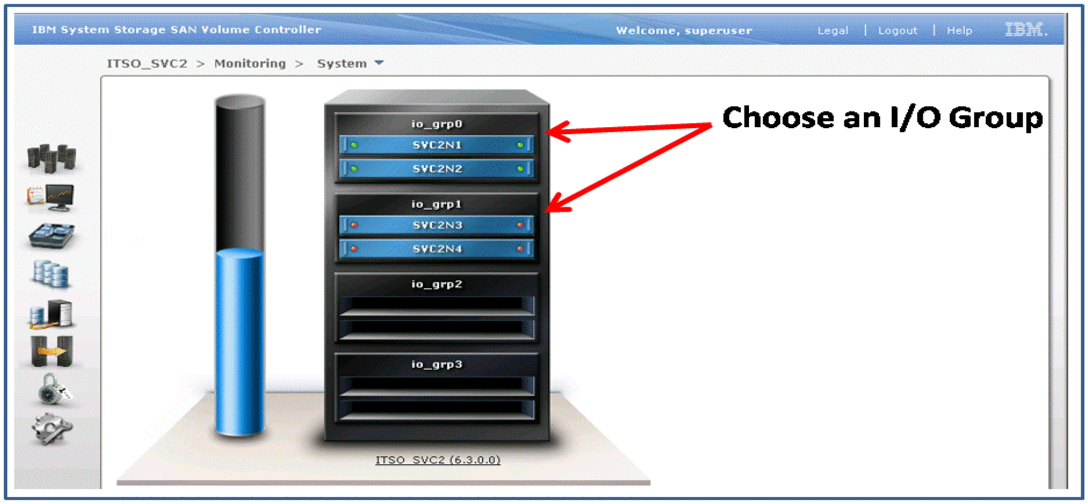


Figure 8-2 Select an I/O Group

You can see how much space is reserved for each feature (FlashCopy, Global and Metro Mirror, Volume Mirroring and RAID), as shown in Figure 8-3.

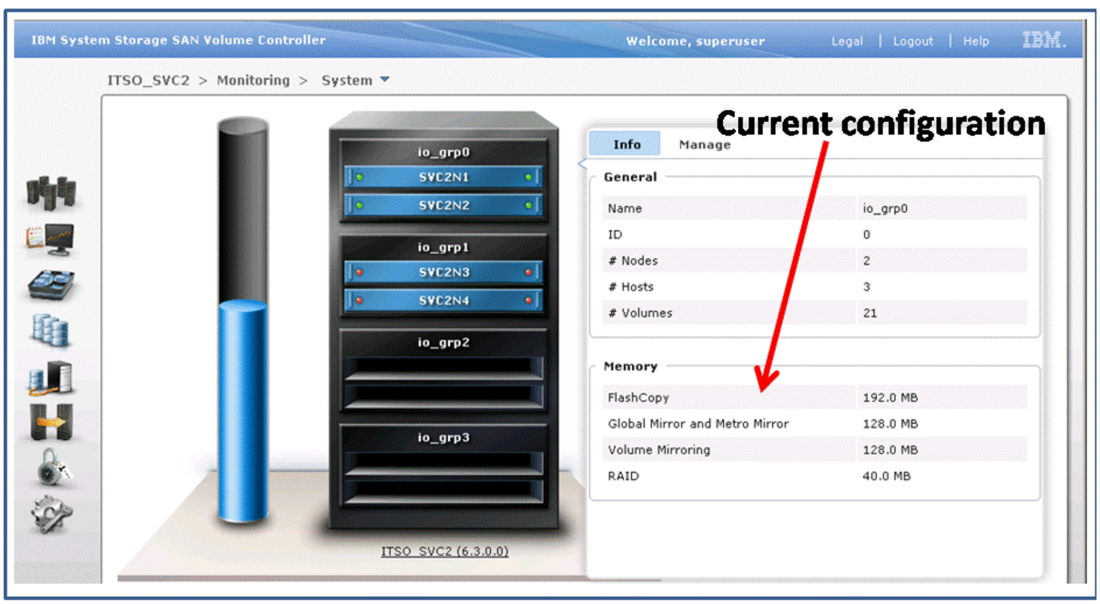


Figure 8-3 View amount of space reserved for each feature

To change the bitmap space capacity for the Volume Mirroring function, click the **Manage** button, change the **Volume Mirroring** field, and then click the **Save** button, as shown in Figure 8-4 on page 302.

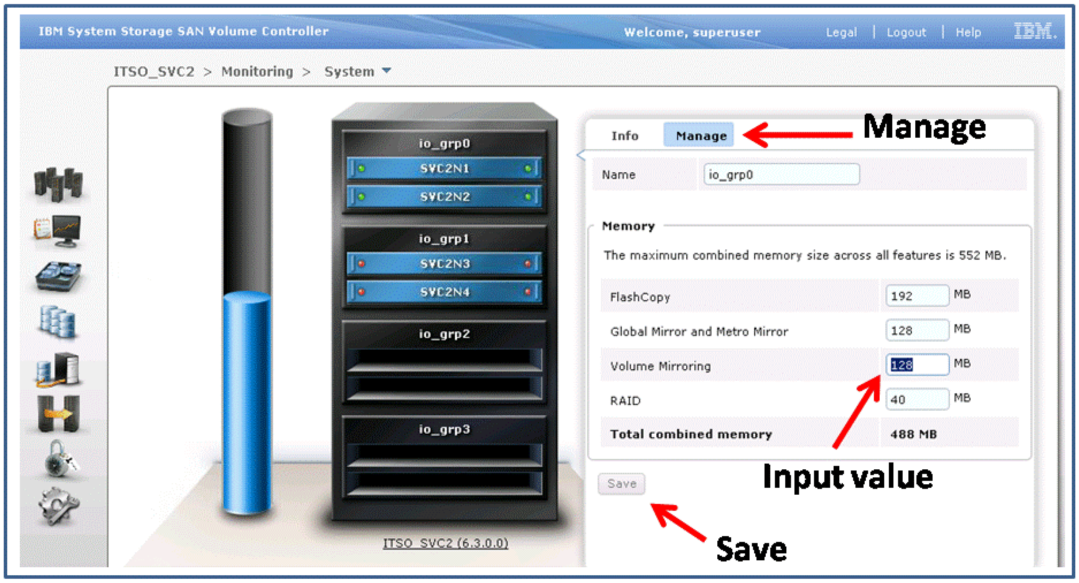


Figure 8-4 Change the bitmap space capacity

8.2 Managing bitmap space for Volume Mirroring using the CLI

Before working with tasks to manage Volume Mirroring, you need to configure bitmap space for Volume Mirroring. In our case, at this point, we are already logged on to the cluster using PuTTy.

> **Automatic bitmap allocations:** SVC 6.2.0 introduced automatic bitmap allocations for creating mirrored volumes using the GUI or CLI (**`mkvdisk`** or **`addvdiskcopy`**).

Each I/O group has an independent bitmap space configuration. The maximum value shared between all features (FlashCopy, Global and Metro Mirror, Volume Mirroring and RAID) is 552 MB. To determine how much space is currently defined for Volume Mirroring, follow these steps:

1. Use the **`lsiogrp`** command to see all the configured I/O Groups in your cluster (Example 8-1).

Example 8-1 lsiogrp

```
IBM_2145:ITSO_SVC2:superuser>lsiogrp
id name            node_count vdisk_count host_count
0  io_grp0         2          21          3
1  io_grp1         2          16          0
2  io_grp2         0          0           0
3  io_grp3         0          0           0
4  recovery_io_grp 0          0           0
```

2. Choose which I/O Group you want to see the current configured values for using the I/O Group id or name (Example 8-2).

Example 8-2 lsiogrp id

```
Using id parameter:

IBM_2145:ITSO_SVC2:superuser>lsiogrp 0
id 0
name io_grp0
node_count 2
vdisk_count 21
host_count 3
flash_copy_total_memory 192.0MB
flash_copy_free_memory 191.9MB
remote_copy_total_memory 128.0MB
remote_copy_free_memory 127.9MB
mirroring_total_memory 128.0MB
mirroring_free_memory 127.9MB
raid_total_memory 40.0MB
raid_free_memory 40.0MB
maintenance no

Using name parameter:

IBM_2145:ITSO_SVC2:superuser>lsiogrp io_grp0
id 0
name io_grp0
node_count 2
vdisk_count 21
host_count 3
flash_copy_total_memory 192.0MB
flash_copy_free_memory 191.9MB
remote_copy_total_memory 128.0MB
remote_copy_free_memory 127.9MB
mirroring_total_memory 128.0MB
mirroring_free_memory 127.9MB
raid_total_memory 40.0MB
raid_free_memory 40.0MB
maintenance no
```

3. To change values, use the **chiogrp** command with **-size** and **-feature** as parameters (Example 8-3).

> **Tip:** Because we want to change the bitmap space for Volume Mirroring, we need to use the attribute **mirror**. To see all acceptable attributes, use the **chiogrp -h** command.

Example 8-3 chiogrp

```
IBM_2145:ITSO_SVC2:superuser>chiogrp -feature mirror -size 140 io_grp0
```

4. To check new configured values, use the **lsiogrp** command (Example 8-4).

Example 8-4 lsiogrp 0

```
IBM_2145:ITSO_SVC2:superuser>lsiogrp 0
id 0
name io_grp0
```

```
node_count 2
vdisk_count 21
host_count 3
flash_copy_total_memory 192.0MB
flash_copy_free_memory 191.9MB
remote_copy_total_memory 128.0MB
remote_copy_free_memory 127.9MB
mirroring_total_memory 140.0MB
mirroring_free_memory 139.9MB
raid_total_memory 40.0MB
raid_free_memory 40.0MB
maintenance no
```

5. At this point, the bitmap space for Volume Mirroring has been changed.

8.3 Configuring Volume Mirroring using the GUI

These sections explain how to perform the following tasks by using the GUI of the SAN Volume Controller (SVC) or Storwize V7000:

- Creating a new volume; see 8.3.1, "Creating a new volume" on page 304.
- Adding a mirrored copy to an existing volume, see 8.3.2, "Adding a mirrored copy to an existing volume" on page 313.
- Deleting a mirrored copy from a volume mirror, see 8.3.3, "Deleting a mirrored copy from a volume mirror" on page 316.
- Splitting a volume copy, see 8.3.4, "Splitting a volume copy" on page 317.
- Validating volume copies, see 8.3.5, "Validating volume copies" on page 318.
- Migrating to a thin-provisioned volume using Volume Mirroring, see 8.3.6, "Migrating to a thin-provisioned volume using Volume Mirroring" on page 319.

8.3.1 Creating a new volume

> **Existing generic volume:** If you already have a generic volume, go to 8.3.2, "Adding a mirrored copy to an existing volume" on page 313.

If you do not already have a generic volume, you first need to create one by following these steps:

1. Go to the All Volumes panel from the SVC/Storwize Welcome panel and click **Volumes** → **All Volumes** as shown in Figure 8-5 on page 305.

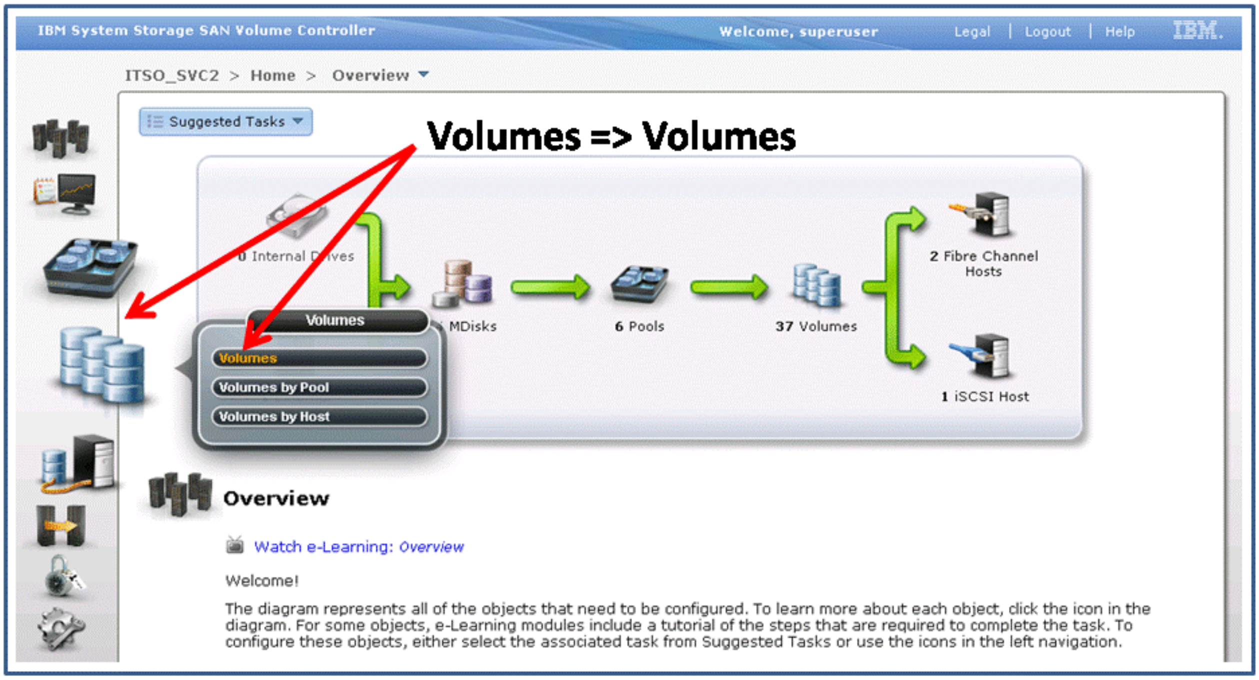
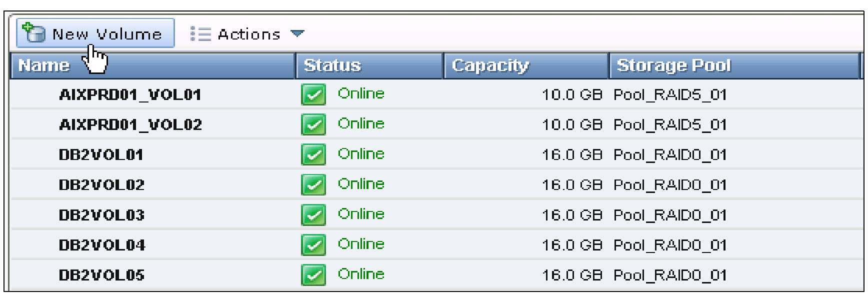


Figure 8-5 Selecting All Volumes

2. Click **New Volume** (Figure 8-6).



Name	Status	Capacity	Storage Pool
AIXPRD01_VOL01	Online	10.0 GB	Pool_RAID5_01
AIXPRD01_VOL02	Online	10.0 GB	Pool_RAID5_01
DB2VOL01	Online	16.0 GB	Pool_RAID0_01
DB2VOL02	Online	16.0 GB	Pool_RAID0_01
DB2VOL03	Online	16.0 GB	Pool_RAID0_01
DB2VOL04	Online	16.0 GB	Pool_RAID0_01
DB2VOL05	Online	16.0 GB	Pool_RAID0_01

Figure 8-6 New Volume action

3. Select one of the following presets, as shown in Figure 8-7 on page 306:

> **Compressed volumes:** At the time of writing, compressed volumes were not available. To learn more about how to configure compressed volumes and IBM Real-time Compression™ refer to *Real-time Compression in SAN Volume Controller and Storwize V7000*, REDP-4859

   – Generic: Create volumes that use a set amount of capacity from the selected storage pool.

   – Thin Provision: Create volumes whose capacity is large, but which only use the capacity that is written by the host application from the pool.

– Mirror: Create volumes with two physical copies that provide data protection. Each copy can belong to a separate storage pool to protect data from storage failures.

– Thin Mirror: Create volumes with two physical copies to protect data from failures while using only the capacity that is written by the host application.

> **Changing the preset:** For our example, we chose the Generic preset. However, whatever selected preset you choose, you can reconsider your decision later and customize the volume by clicking the Advanced option.

4. After selecting a preset (in our example, Generic), you must select the storage pool on which the data will be striped (Figure 8-7).

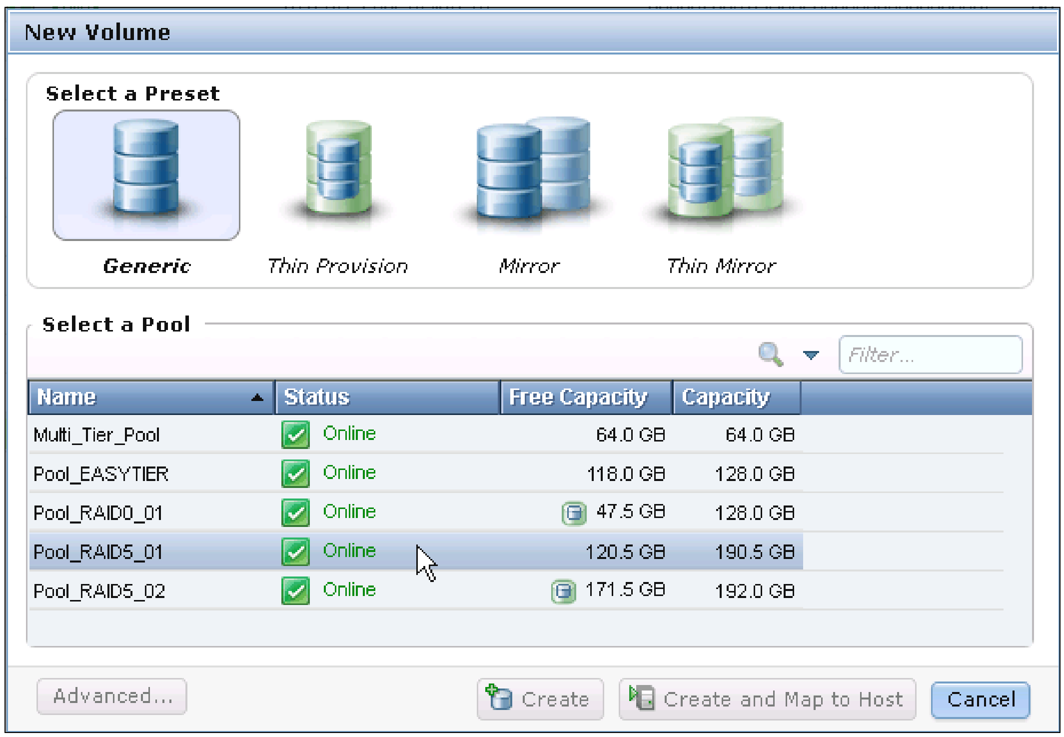


Figure 8-7 Select the storage pool on which the data will be striped

5. After you select the storage pool, the window will be updated automatically. You must select a volume name and size, as shown in Figure 8-8 on page 307:

– Enter a name if you want to create a single volume, or a naming prefix if you want to create multiple volumes.

> **Volume name:** You can use the letters A to Z and a to z, the numbers 0 to 9, and the underscore (_) character. The host name can be between one and 63 characters in length.

– Enter the size of the volume that you want to create and select the capacity unit of measurement (bytes, KB, MB, GB, or TB) from the list.

> **Tip:** An entry of 1 GB uses 1024 MB.

An updated summary automatically appears in the bottom of the window to show the amount of space that will be used and the amount of free space that remains in the pool.

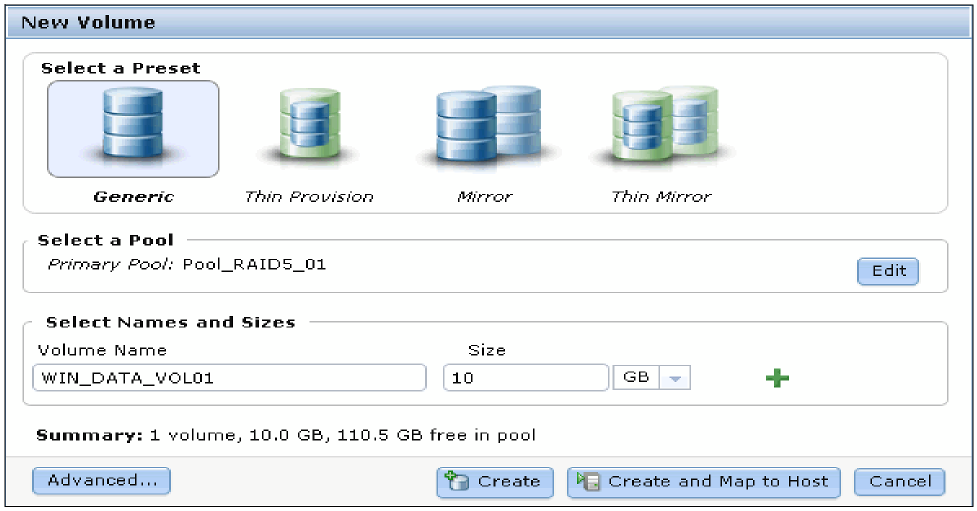


Figure 8-8 New Volume: Select Names and Sizes

Various optional actions are available from this window:

– You can modify the storage pool by clicking **Edit**. In this case, you can select another storage pool.

– You can create additional volumes by clicking the green plus sign  icon. You can repeat this action as many times as necessary. You can remove the volumes by clicking the red  icon.

> **Naming:** When you create more than one volume, the wizard does not ask you for a name for each volume to be created. Instead, the name that you use here becomes the prefix and a number, starting at zero (0), is appended to this prefix as each volume is created.

6. You can activate and customize advanced features, such as thin provisioning or mirroring, depending on the preset that you have selected. To access these settings, click **Advanced**:

On the Characteristics tab (Figure 8-9 on page 308), you can set the following options:

– General: Format the new volume by selecting the **Format Before Use** check box. (Formatting writes zeros to the volume before it can be used; that is, it writes zeros to its MDisk extents.)

– Locality: Choose an I/O Group and then select a preferred node.

– OpenVMS only: Enter the user-defined identifier (UDID) for OpenVMS. You only need to complete this field for the OpenVMS system.

> **UDID:** Each OpenVMS fibre-attached volume requires a user-defined identifier or unit device identifier (UDID). A UDID is a non-negative integer that is used when an OpenVMS device name is created. To recognize volumes, OpenVMS issues a UDID value, which is a unique numerical number.

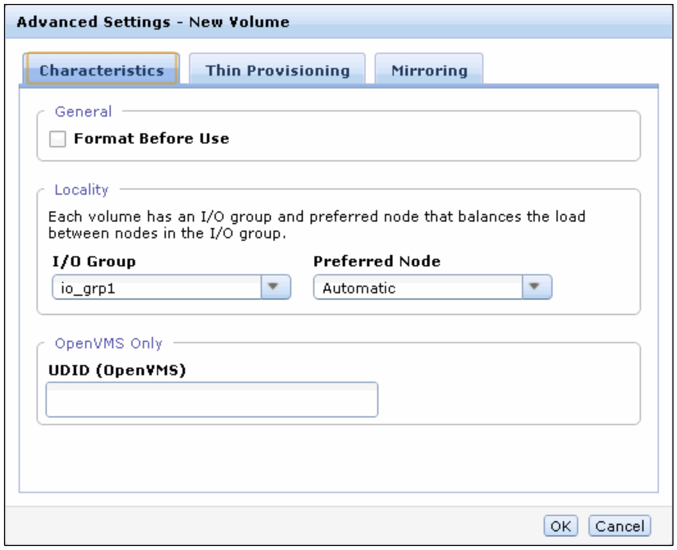


Figure 8-9 Advanced Settings: Characteristics

On the Thin Provisioning tab (Figure 8-10 on page 309), after you activate thin provisioning by selecting the **Enable Thin Provisioning** check box, you can set the following options:

– Real: Type the real size that you want to allocate. This size is the amount of disk space that actually will be allocated. It either can be a percentage of the virtual size or a specific number in GBs.

– Automatically Expand: Select auto expand, which allows the real disk size to grow, as required.

– Warning Threshold: Type a percentage or select a specific size for the usage threshold warning. It will generate a warning when the used disk capacity on the space-efficient copy first exceeds the specified threshold.

– Thin-Provisioned Grain Size: Select the grain size: 32 KB, 64 KB, 128 KB, or 256 KB. Smaller grain sizes save space, and larger grain sizes produce better performance. Try to match the FlashCopy grain size if the volume will be used for FlashCopy.

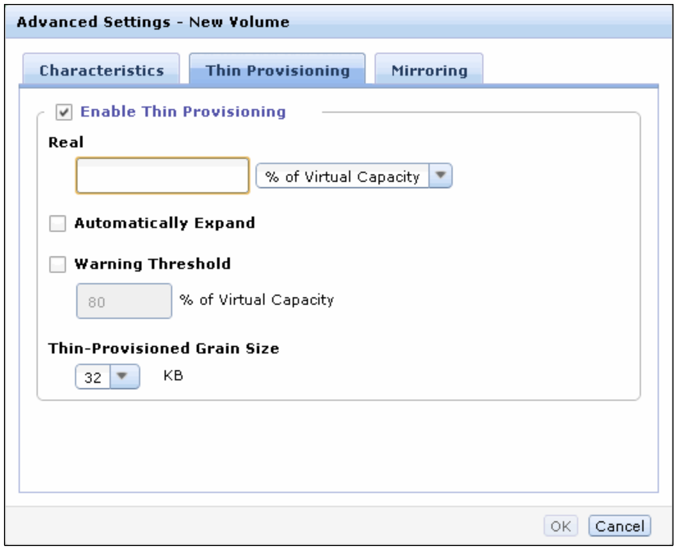


Figure 8-10 Advanced Settings: Thin Provisioning

> **Important:** If you selected the Thin Provision or Thin Mirror preset on the first page (Figure 8-8 on page 307), the Enable Thin Provisioning check box is already selected and the following parameter preset values are pre-filled:
>
> - Real: 2% of Virtual Capacity
> - Automatically Expand: Selected
> - Warning Threshold: Selected with a value of 80% of Virtual Capacity
> - Thin-Provisioned Grain Size: 32 KB

On the Mirroring tab (Figure 8-11 on page 310), after you activate mirroring by selecting the **Create Mirrored Copy** check box, you can set the following option:

– Mirror Sync Rate: Enter the *Mirror Synchronization rate*. It is the I/O governing rate in a percentage that determines how quickly copies are synchronized. A zero value disables synchronization.

> **Important:** If you activate this feature from the Advanced menu, you must select a secondary pool on the main window (Figure 8-8 on page 307).
>
> The primary pool will be used as the primary and preferred copy for read operations. The secondary pool will be used as the secondary copy.

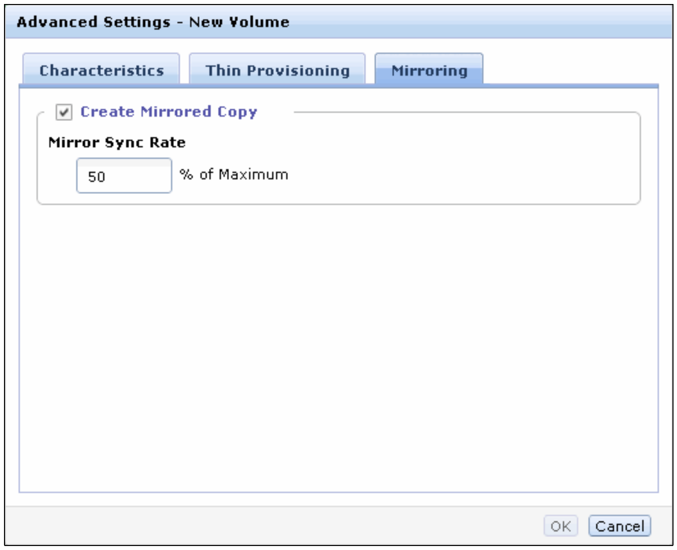


Figure 8-11 Advanced Settings: Mirroring

> **Important:** If you selected the Mirror or Thin Mirror preset on the first page (Figure 8-8 on page 307), the Create Mirrored Copy check box is already selected and the Mirror Sync Rate preset is pre-filled with 80% of Maximum.

7. After you have set all of the advanced settings, click **OK** to return to the main menu (Figure 8-8 on page 307).
8. Then, you can choose to create only the volume by clicking Create, or to create and map the volume by selecting “Create and Map to Host”:

   If you select to create only the volume, you will return to the main All Volumes panel. You see that your volume has been created but not mapped (Figure 8-12). You can map it later.

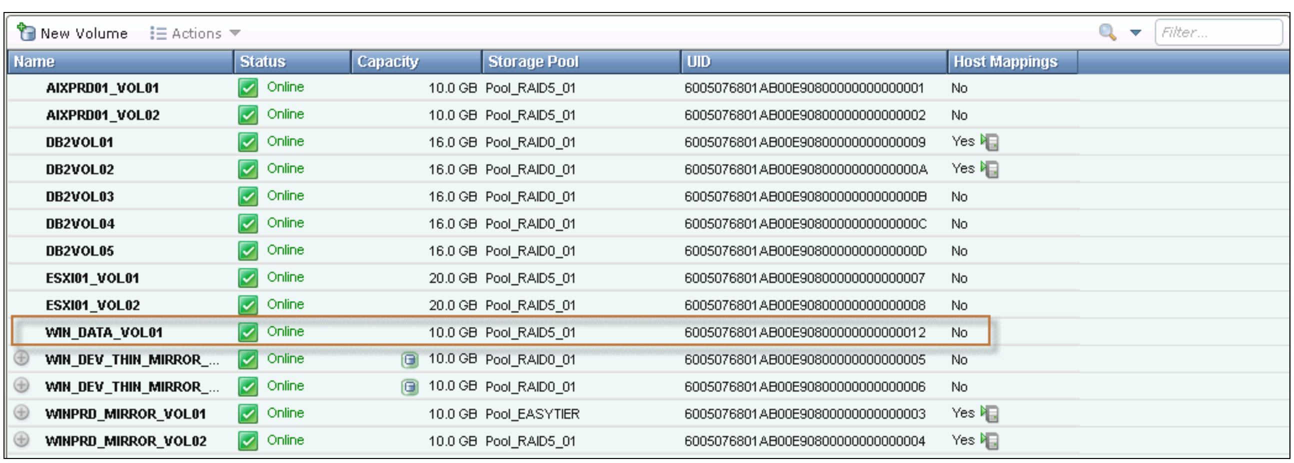


Figure 8-12 Volume created without mapping

Or, if you want to create and map the volume on the volume creation window, click **Continue** and another window opens. In the Modify Mappings window, select the host to which you want to map this volume by using the drop-down list box, and then click **Next** (Figure 8-13 on page 311).

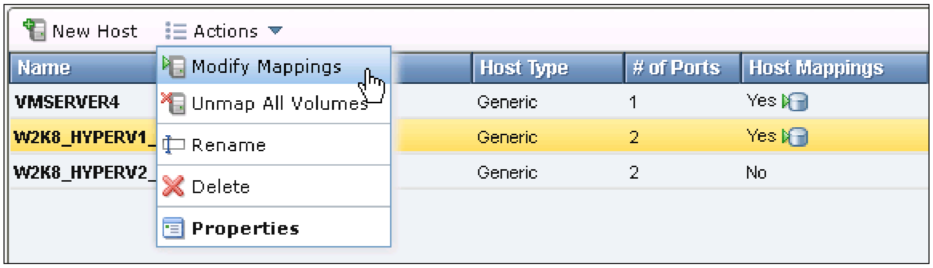


Figure 8-13 Select the host to which to map your volume

In the Modify Host Mappings window, verify the mapping. If you want to modify the mapping, select the volume or volumes that you want to map to a host and move each of them to the table on the right by using the right (>) arrow, as shown in Figure 8-14. If you need to remove the mappings, use the left arrow.

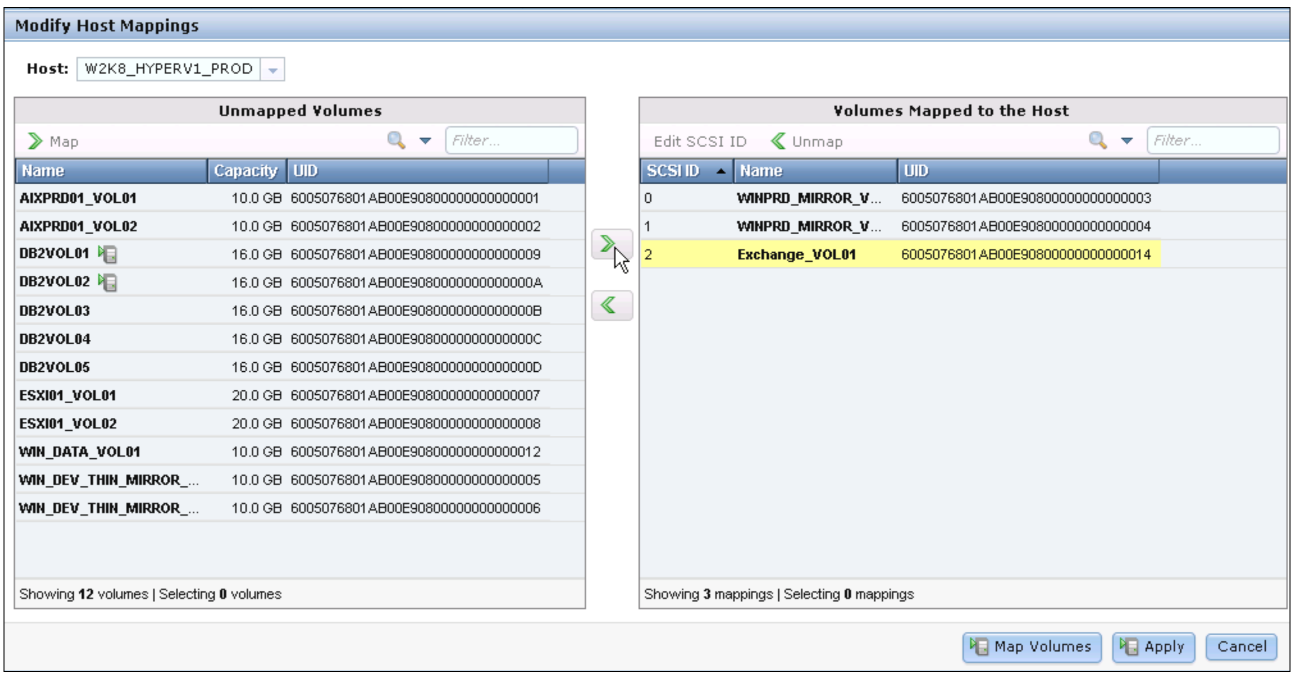


Figure 8-14 Modify Host Mappings window: Adding volumes to a host

In the table on the right, you can edit the SCSI ID by selecting a mapping that is highlighted in yellow, which indicates a new mapping. Next, click **Edit SCSI ID**, as shown in Figure 8-14.

> **Changing the SCSI ID:** You can only change the SCSI ID on new mappings. To edit an existing mapping's SCSI ID, you must unmap the volume and recreate the map to the volume.

In the Edit SCSI ID window, change the SCSI ID and then click **OK** (Figure 8-15 on page 312).

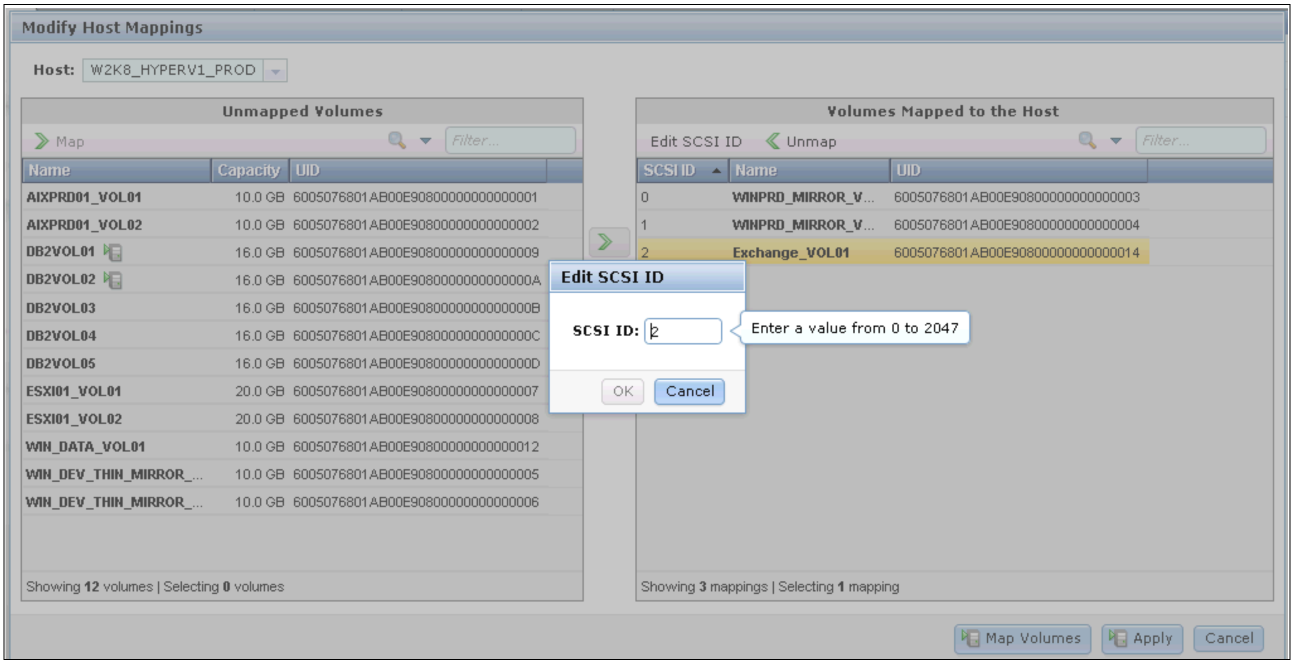


Figure 8-15 Modify Host Mappings window: Edit SCSI ID

After you have added all of the volumes that you want to map to this host, click **Map Volumes** or **Apply** to create the host mapping relationships and finalize the volume creation. You return to the main All Volumes window. Your volume will display as having been created and mapped, as shown in Figure 8-16.

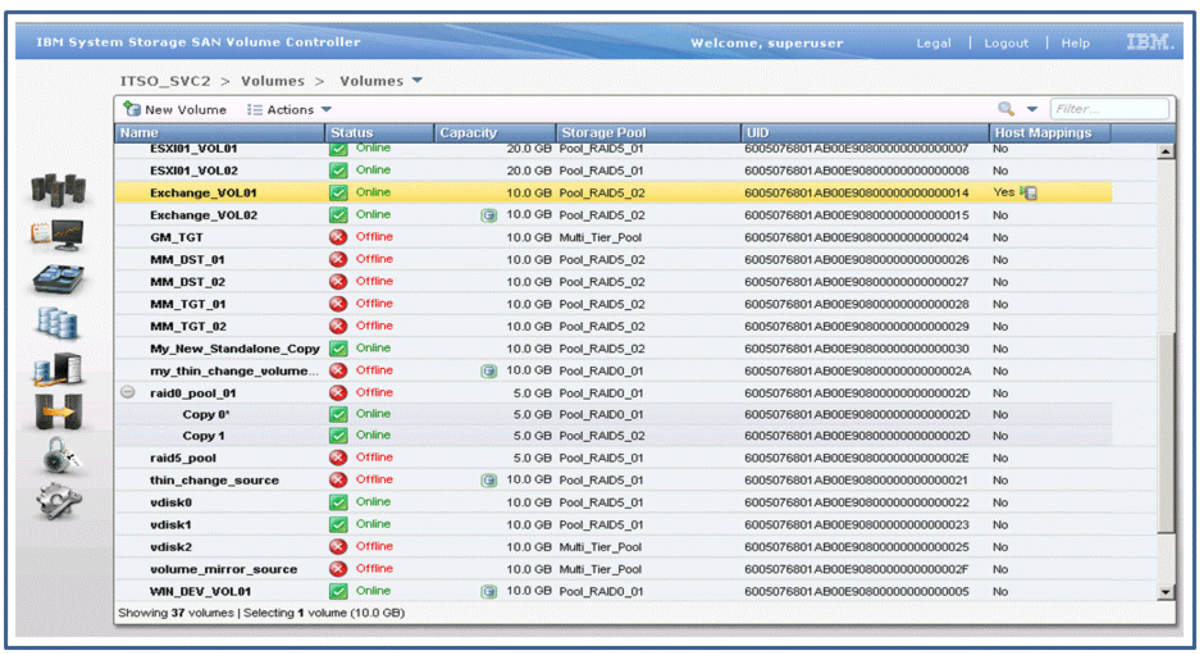


Figure 8-16 Volume has been created with mapping

8.3.2 Adding a mirrored copy to an existing volume

You can add a mirrored copy to an existing volume, which will give you two copies of the underlying disk extents.

Tip: You can also create a new mirrored volume by selecting the Mirror or Thin Mirror preset during the volume creation, as shown in Figure 8-7 on page 306.

You can use a volume mirror for any operation for which you can use a volume. It is transparent to higher-level operations, such as Metro Mirror, Global Mirror, or FlashCopy.

Creating a volume mirror from an existing volume is not restricted to the same storage pool, so it is an ideal method to use to protect your data from a disk system or an array failure. If one copy of the mirror fails, it provides continuous data access to the other copy. When the failed copy is repaired, the copies automatically resynchronize.

You can also use a volume mirror as an alternative migration tool, where you can synchronize the mirror before splitting off the original side of the mirror. The volume stays online, and it can be used normally, while the data is being synchronized. The copies can also be separate structures, that is, striped, image, sequential, or space-efficient, and separate extent sizes.

To create a mirror copy from within a volume, perform the following steps:

1. Select the volume in the table.

2. Click **Actions** → **Volume Copy Actions** → **Add Mirrored Copy** (Figure 8-17 on page 314).

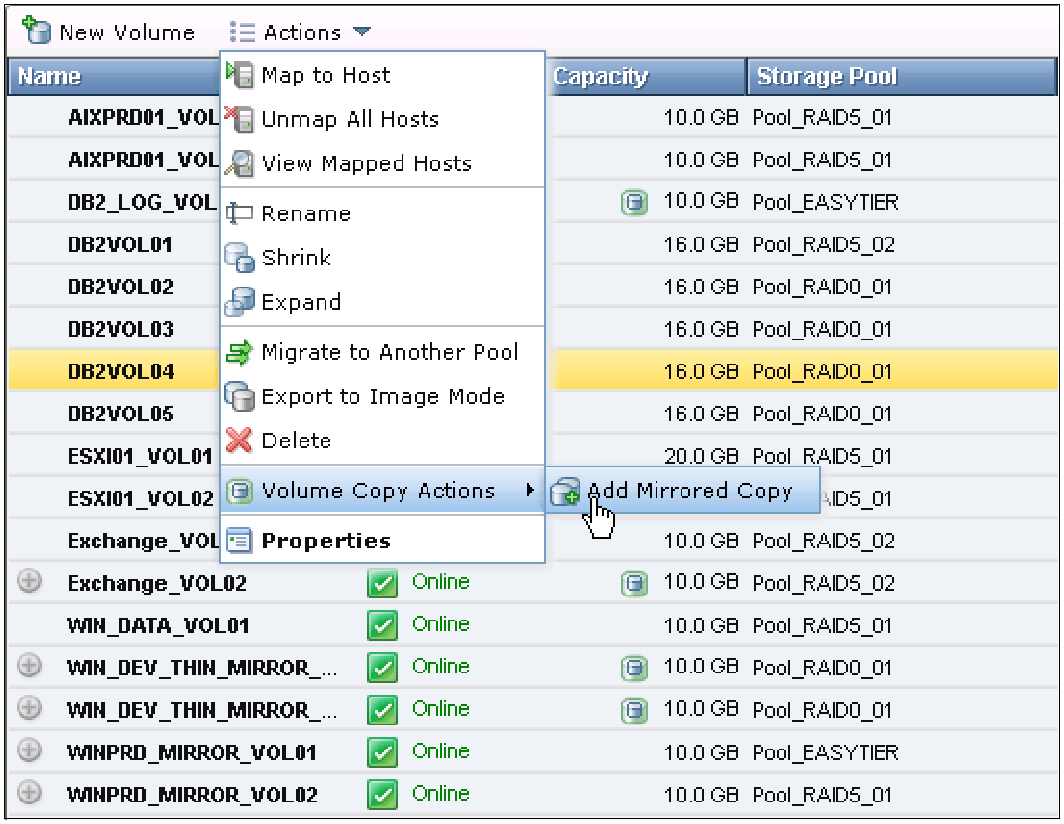


Figure 8-17 Add Mirrored Copy actions

Tip: You can also right-click a volume and select **Volume Copy Actions** → **Add Mirrored Copy** from the list.

3. The Add Volume Copy - *volumename* window (where *volumename* is the volume that you selected in the previous step) opens (2). You can perform the following steps separately or in combination:

   – Select the storage pool in which you want to put the copy. To maintain higher availability, choose a separate group.

   – Select the **Enable Thin Provisioning** check box to make the copy space-efficient.

   The following parameters are used for this thin-provisioned copy:

   - Real Size: 2% of Virtual Capacity
   - Enable Autoexpand: Active
   - Warning Threshold: 80% of Virtual Capacity
   - Thin-Provisioned Grain Size: 32 KB

   Changing options: You can only change Real Size, Enable Autoexpand, and Warning Threshold after the thin-provisioned volume copy has been added.

4. Click **Add Copy** (Figure 8-18 on page 315).

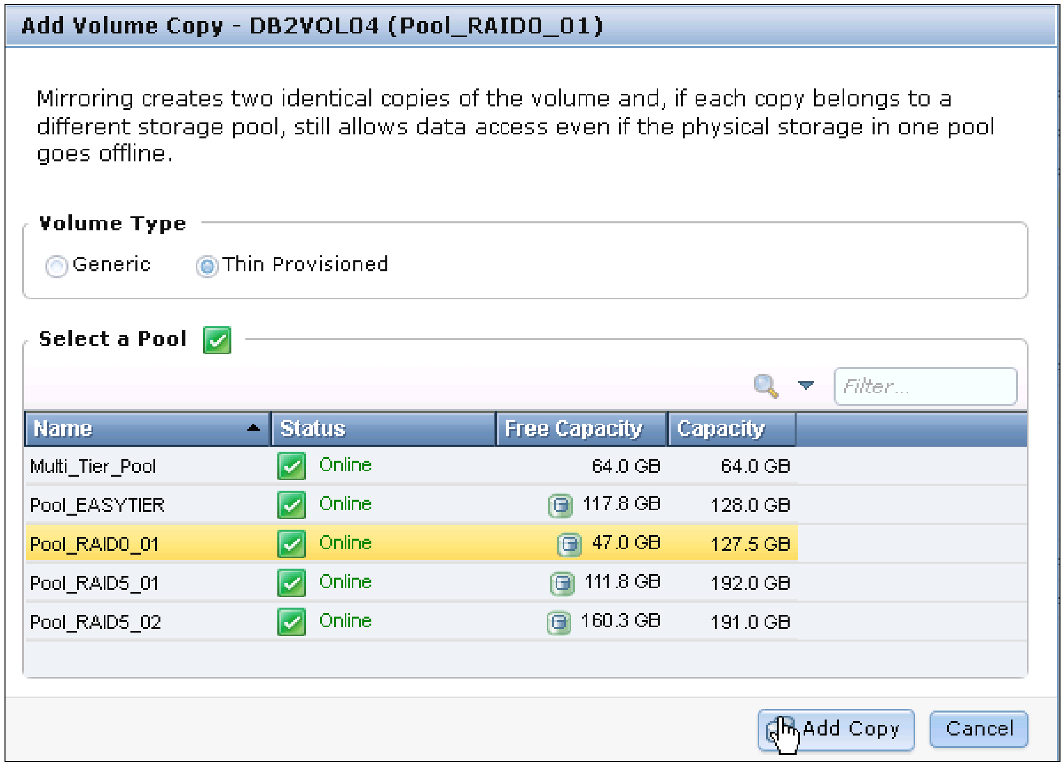


Figure 8-18 Add Copy to volume window

5. You can check the migration using the Running Tasks menu (see Figure 8-19).

   To expand this Status Area, click the icon and click **Volume Synchronization**. Figure 8-19 shows a detailed view of the running tasks.

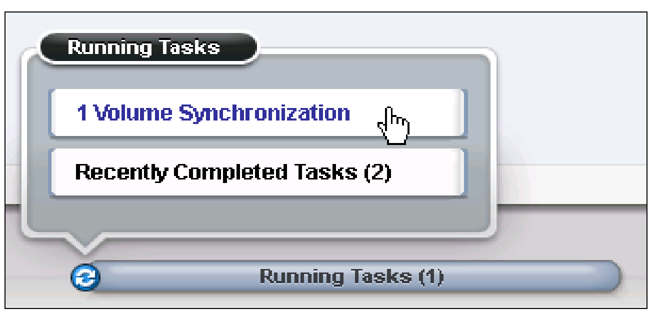
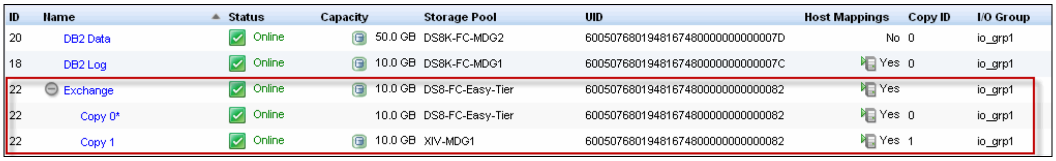


Figure 8-19 Running Task: Volume Synchronization

Mirror Sync rate: You can change the Mirror Sync Rate (the default is 50%) by modifying the volume properties.

6. When synchronization is finished, the volume will be part of the new pool (Figure 8-20 on page 316).



ID	Name	Status	Capacity	Storage Pool	UID	Host Mappings	Copy ID	I/O Group
20	DB2 Data	Online	50.0 GB	DS8K-FC-MDG2	600507680194816748000000000000007D	No	0	io_grp1
18	DB2 Log	Online	10.0 GB	DS8K-FC-MDG1	600507680194816748000000000000007C	Yes	0	io_grp1
22	Exchange	Online	10.0 GB	DS8-FC-Easy-Tier	6005076801948167480000000000000082	Yes		io_grp1
22	Copy 0*	Online	10.0 GB	DS8-FC-Easy-Tier	6005076801948167480000000000000082	Yes	0	io_grp1
22	Copy 1	Online	10.0 GB	XIV-MDG1	6005076801948167480000000000000082	Yes	1	io_grp1

Figure 8-20 Mirrored volume

> **Primary copy:** As shown in Figure 8-20, the primary copy is identified with an asterisk (*). In this example, `Copy 0` is the primary copy.

8.3.3 Deleting a mirrored copy from a volume mirror

To remove a volume copy, perform the following steps:

1. Select the volume copy that you want to remove in the table, and click **Actions** → **Delete this Copy** (Figure 8-21).

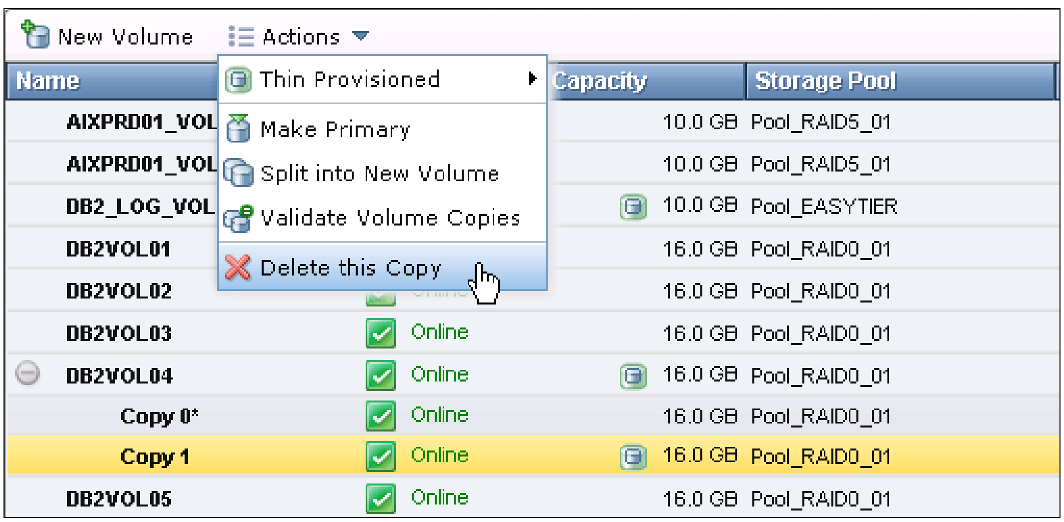


Figure 8-21 Delete this Copy action

> **Tip:** You can also right-click a volume and select Delete this Copy from the list.

2. The Warning window opens (Figure 8-22 on page 317). Click **OK** to confirm your choice.

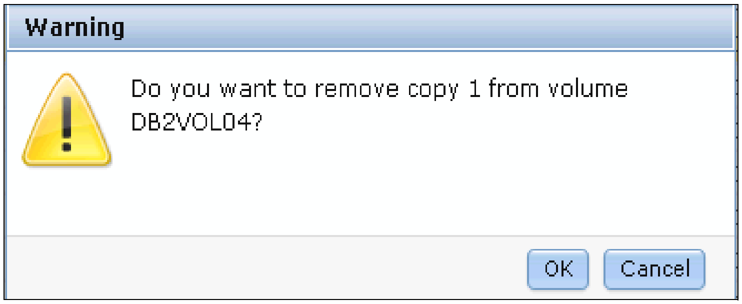


Figure 8-22 Warning window

> **Removing a primary copy:** If you try to remove the primary copy before it has been synchronized with the other copy, you will receive the message: `The command failed because the copy specified is the only synchronized copy`. You must wait until the end of the synchronization to be able to remove this copy.

3. The copy is now deleted.

8.3.4 Splitting a volume copy

To split off a synchronized volume copy to a new volume, perform the following steps:

1. In the table, select the volume copy that you want to split, and click **Actions** → **Split into New Volume** (Figure 8-23).

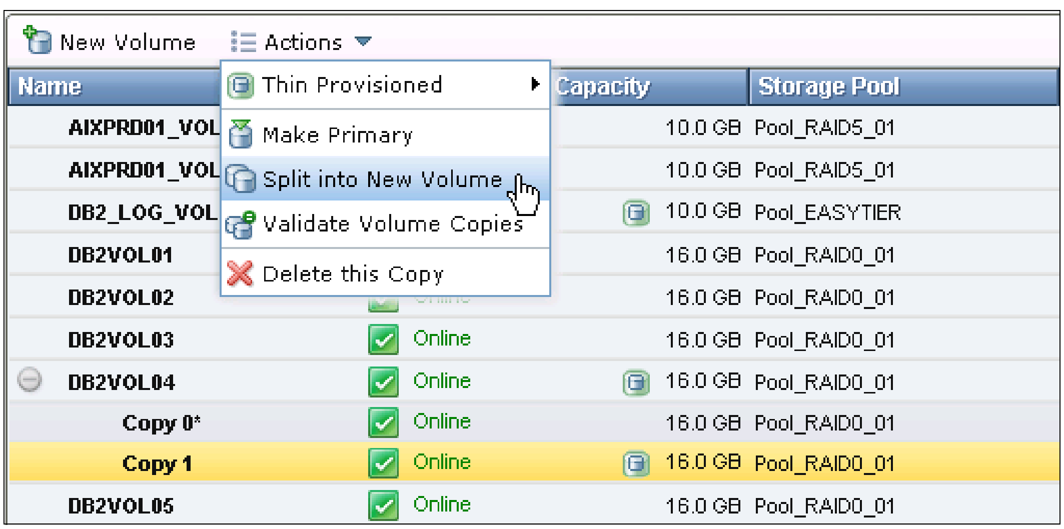


Figure 8-23 Split into New Volume action

> **Tip:** You can also right-click a volume and select Split into New Volume from the list.

2. The Split Volume Copy window opens (Figure 8-24 on page 318). In this window, type a name for the new volume.

Volume name: If you do not provide a name, the cluster automatically generates the name volumeX (where *x* is the ID sequence number that is assigned by the cluster internally).

If you want to provide a name, you can use the letters A to Z and a to z, the numbers 0 to 9, and the underscore (_). The host name can be between one and 63 characters in length.

3. Click **Split Volume Copy** (Figure 8-24).

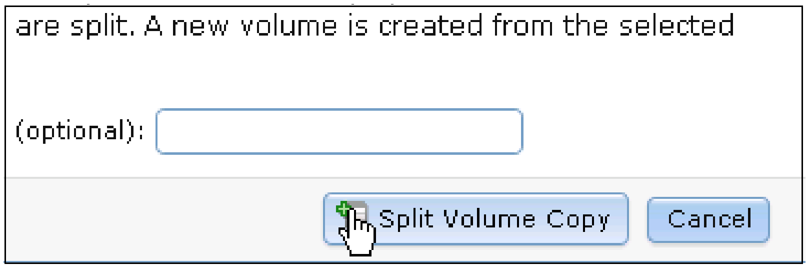


Figure 8-24 Split Volume Copy window

4. This new volume is now available to be mapped to a host.

Important: After you split a volume mirror, you cannot resynchronize or recombine them and you must create a new volume copy.

8.3.5 Validating volume copies

To validate the copies of a mirrored volume, perform the following steps:

1. Select a copy of this volume in the table, and click **Actions** → **Validate Volume Copies** (Figure 8-25).

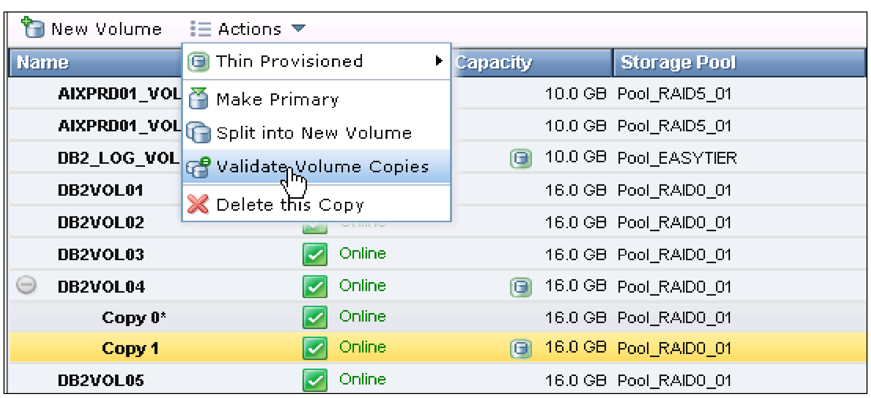


Figure 8-25 Validate Volume Copies actions

2. The Validate Volume Copies window opens (Figure 8-26 on page 319). In this window, select one of the following options:

   – Generate Event of Differences: Use this option if you only want to verify that the mirrored volume copies are identical. If a difference is found, the command stops and logs an error that includes the logical block address (LBA) and the length of the first difference.

You can use this option, starting at a separate LBA each time, to count the number of differences on a volume.

– Overwrite Differences: Use this option to overwrite the content from the primary volume copy to the other volume copy. The command corrects any differing sectors by copying the sectors from the primary copy to the copies being compared. Upon completion, the command process logs an event, which indicates the number of differences that were corrected.

  Use this option if you are sure that either the primary volume copy data is correct, or that your host applications can handle incorrect data.

– Return Media Error to Host: Use this option to convert sectors on all volume copies that contain different contents into virtual medium errors. Upon completion, the command logs an event, which indicates the number of differences that were found, the number of differences that were converted into medium errors, and the number of differences that were not converted.

  Use this option if you are unsure what the correct data is, and you do not want an incorrect version of the data to be used.

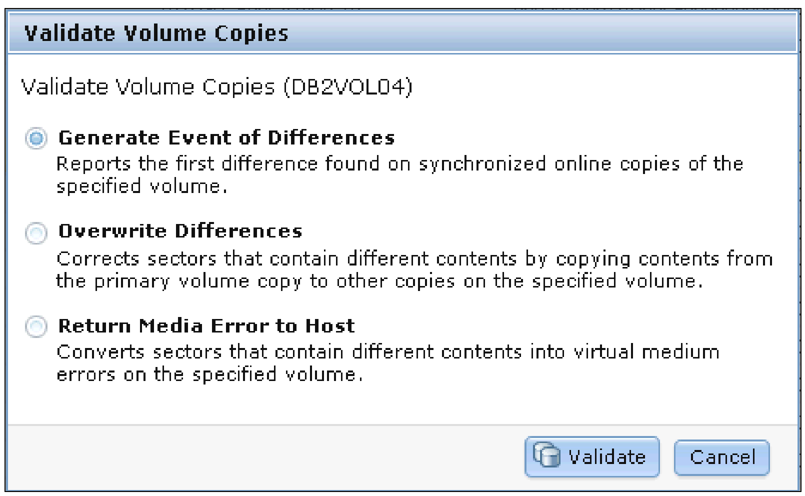


Figure 8-26 Validate Volume Copies

3. Click **Validate** (Figure 8-26).
4. The volume is now checked.

8.3.6 Migrating to a thin-provisioned volume using Volume Mirroring

To migrate to a thin-provisioned volume, perform the following steps:

1. Select the volume in the table.

2. Click **Actions** → **Volume Copy Actions** → **Add Mirrored Copy** (Figure 8-27 on page 320).

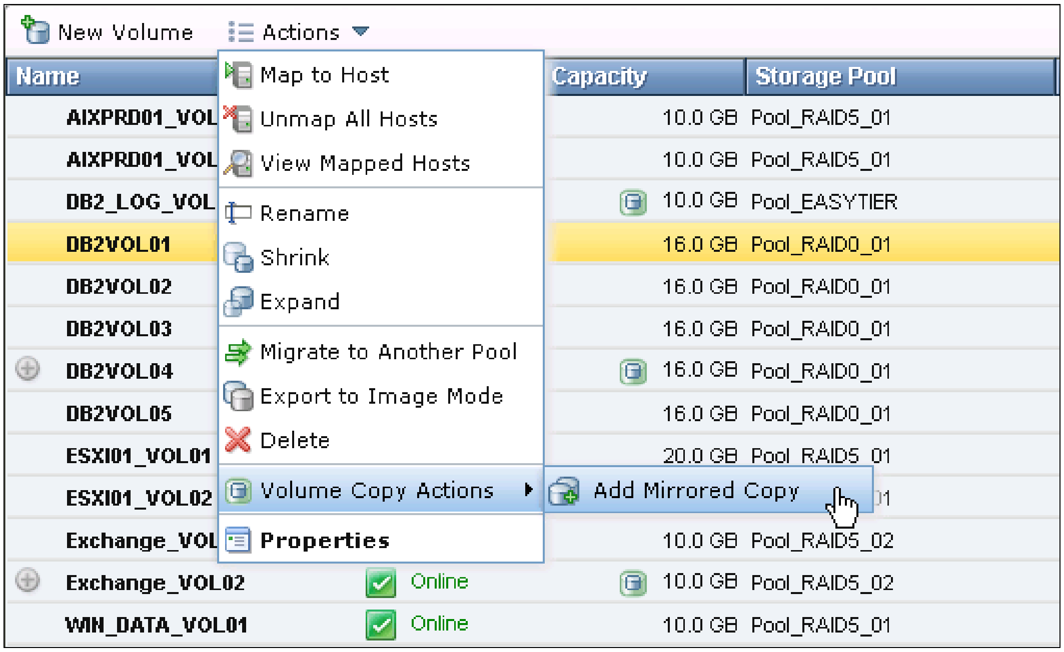


Figure 8-27 Add Mirrored Copy actions

> **Tip:** You can also right-click a volume, select **Volume Copy Actions** → **Add Mirrored Copy**.

3. The Add Volume Copy - *volumename* window (where *volumename* is the volume that you selected in the previous step) opens (Figure 8-28 on page 321). You can perform the following steps separately or in combination:

   – Select the storage pool in which you want to put the copy. To maintain higher availability, choose a separate group.

   – Select **Enable Thin Provisioning** to make the copy space-efficient.

   The following parameters are used for this thin-provisioned copy:

   - Real Size: 2% of Virtual Capacity
   - Autoexpand: Active
   - Warning Threshold: 80% of Virtual Capacity
   - Thin-Provisioned Grain Size: 32 KB

   > **Changing options:** You can change the Real Size, Autoexpand, and Warning Threshold after the volume copy has been added in the GUI. For the Thin-Provisioned Grain Size, you need to use the command-line interface (CLI).

4. Click **Add Copy**.

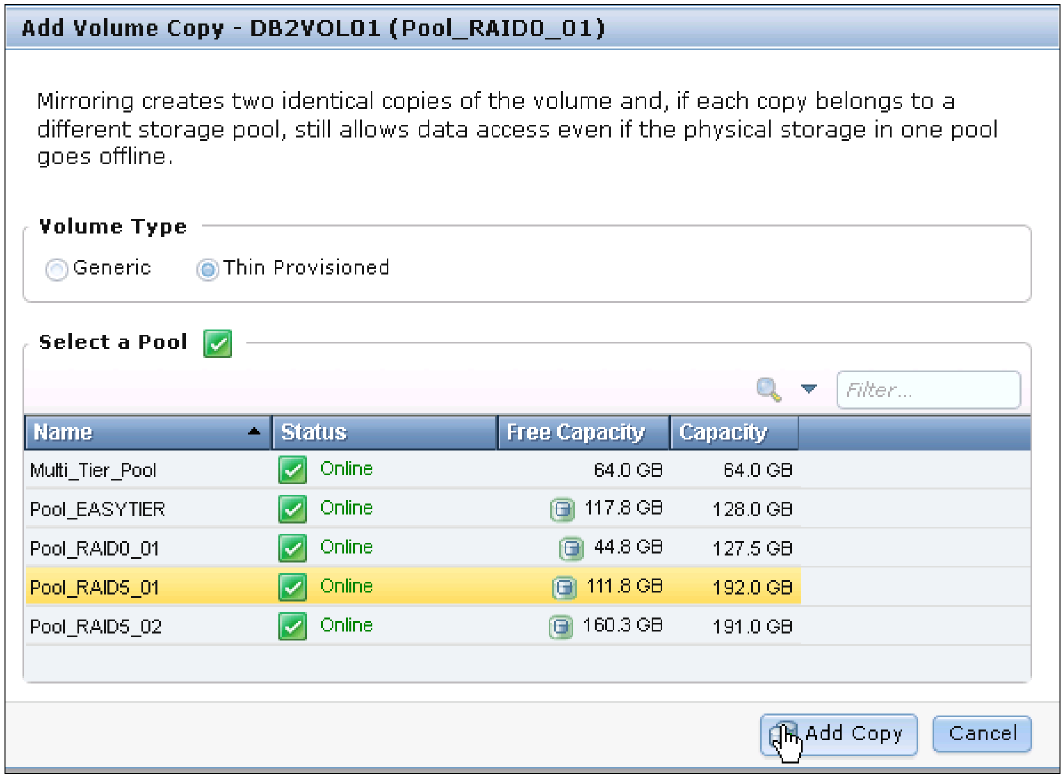


Figure 8-28 Add Volume Copy window

5. You can check the migration by using the Running Tasks status area menu.

   To expand this status area, click the icon and click **Volume Synchronization**. Figure 8-29 shows the detailed view of the running tasks.

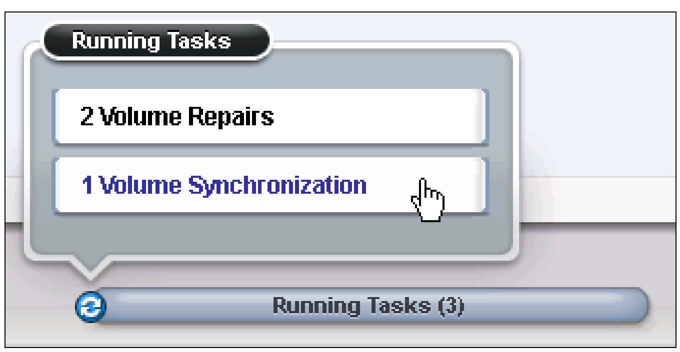


Figure 8-29 Running Tasks status area: Volume Synchronization

Mirror Sync Rate: You can change the Mirror Sync Rate (by default at 50%) by modifying the volume properties.

6. When the synchronization is finished, select the non-thin-provisioned copy that you want to remove in the table. Select **Actions** → **Delete this Copy** (Figure 6).

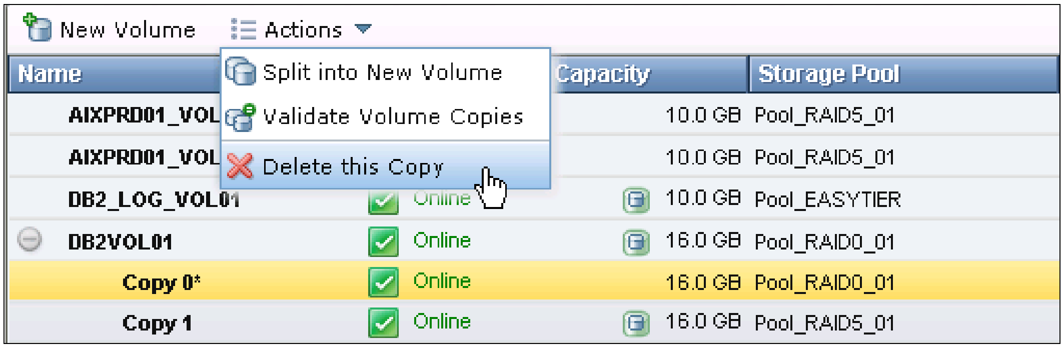


Figure 8-30 Delete this Copy window

> **Tip:** You can also right-click a volume and select Delete this Copy from the list.

7. The Warning window opens (Figure 8-31). Click **OK** to confirm your choice.

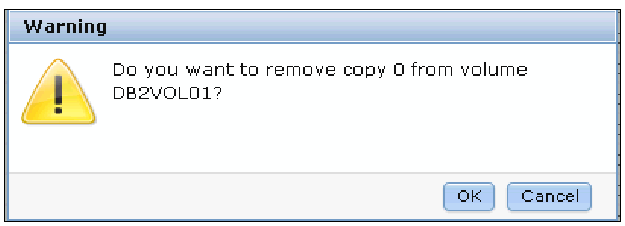


Figure 8-31 Warning window

> **Tip:** If you try to remove the primary copy before it has been synchronized with the other copy, you will receive the following message: `The command failed because the copy specified is the only synchronized copy`. You must wait until the end of the synchronization to be able to remove this copy.

8. When the copy is deleted, your thin-provisioned volume is ready for use.

8.4 Configuring Volume Mirroring using the CLI

These sections explain how to perform the following tasks by using the CLI of the SAN Volume Controller (SVC) or Storwize V7000:

- Creating volumes, see 8.4.1, “Creating a new volume” on page 323.
- Adding a mirrored copy to an existing volume, see 8.4.2, “Adding a mirrored copy to an existing volume” on page 324.
- Deleting a mirrored copy from a volume mirror, see 8.4.3, “Deleting a mirrored volume” on page 328.
- Splitting a volume copy, see 8.4.4, “Splitting a mirrored volume” on page 329.
- Validating volume copies, see 8.4.5, “Validating a mirrored volume” on page 330.

- Configuring I/O Time-out, see 8.4.6, “Configuring I/O Time-out for Mirrored Volumes” on page 331.

8.4.1 Creating a new volume

The **mkvdisk** command creates sequential, striped, or image mode volume objects. When they are mapped to a host object, these objects are seen as disk drives with which the host can perform I/O operations.

When creating a volume, you must enter several parameters at the CLI. There are both mandatory and optional parameters.

> **Creating an image mode disk:** If you do not specify the **-size** parameter when you create an image mode disk, the entire MDisk capacity is used.

When you are ready to create a volume, you must know the following information before you start creating the volume:

- In which storage pool the volume is going to have its extents
- From which I/O Group the volume will be accessed
- Which cluster node will be the preferred node for the volume
- Size of the volume
- Name of the volume
- Type of the volume
- Whether this volume will be managed by Easy Tier to optimize its performance

When you are ready to create your striped volume, use the **mkvdisk** command. In Example 8-5, this command creates a 10 GB striped volume with volume ID 20 within the storage pool `STGPool_DS3500-2` and assigns it to the `iogrp_0` I/O Group. Its preferred node will be node `1`.

Example 8-5 mkvdisk command

```
IBM_2145:ITSO_SVC1:admin>mkvdisk -mdiskgrp STGPool_DS3500-2 -iogrp io_grp0 -node 1
-size 10 -unit gb -name Tiger
Virtual Disk, id [20], successfully created
```

To verify the results, use the **lsvdisk** command, as shown in Example 8-6.

Example 8-6 lsvdisk command

```
IBM_2145:ITSO_SVC1:admin>lsvdisk 20
id 20
name Tiger
IO_group_id 0
IO_group_name io_grp0
status online
mdisk_grp_id 1
mdisk_grp_name STGPool_DS3500-2
capacity 10.00GB
type striped
formatted no
mdisk_id
mdisk_name
FC_id
FC_name
RC_id
```

```
RC_name
vdisk_UID 6005076801AF813F1000000000000016
throttling 0
preferred_node_id 1
fast_write_state empty
cache readwrite
udid
fc_map_count 0
sync_rate 50
copy_count 1
se_copy_count 0
filesystem
mirror_write_priority latency

copy_id 0
status online
sync yes
primary yes
mdisk_grp_id 1
mdisk_grp_name STGPool_DS3500-2
type striped
mdisk_id
mdisk_name
fast_write_state empty
used_capacity 10.00GB
real_capacity 10.00GB
free_capacity 0.00MB
overallocation 100
autoexpand
warning
grainsize
se_copy no
easy_tier on
easy_tier_status inactive
tier generic_ssd
tier_capacity 0.00MB
tier generic_hdd
tier_capacity 10.00GB
```

At this point, you have completed the required tasks to create a volume.

8.4.2 Adding a mirrored copy to an existing volume

You can create a mirrored copy of a volume, which keeps a volume accessible even when the MDisk on which it depends has become unavailable. You can create a copy of a volume either on separate storage pools or by creating an image mode copy of the volume. Copies increase the availability of data; however, they are not separate objects. You can only add, remove, or split copies from the volume.

In addition, you can use Volume Mirroring as an alternative method of migrating volumes between storage pools.For example, if you have a non-mirrored volume in one storage pool and want to migrate that volume to another storage pool, you can add a new copy of the volume and specify the second storage pool. After the copies are synchronized, you can delete the copy on the first storage pool. The volume is copied to the second storage pool while remaining online during the copy. To create a mirrored copy of a volume, use the **addvdiskcopy** command. This command adds a copy of the chosen volume to the selected storage pool, which changes a non-mirrored volume into a mirrored volume.

In the following scenario, we show creating a mirrored volume from one storage pool to another storage pool. Example 8-7 on page 325 shows the volume has a copy with copy_id 0.

Example 8-7 lsvdisk

```
IBM_2145:ITSO_SVC1:admin>lsvdisk Volume_no_mirror
id 23
name Volume_no_mirror
IO_group_id 0
IO_group_name io_grp0
status online
mdisk_grp_id 0
mdisk_grp_name STGPool_DS3500-1
capacity 1.00GB
type striped
formatted no
mdisk_id
mdisk_name
FC_id
FC_name
RC_id
RC_name
vdisk_UID 6005076801AF813F1000000000000019
throttling 0
preferred_node_id 1
fast_write_state empty
cache readwrite
udid
fc_map_count 0
sync_rate 50
copy_count 1
se_copy_count 0
filesystem
mirror_write_priority latency

copy_id 0
status online
sync yes
primary yes
mdisk_grp_id 0
mdisk_grp_name STGPool_DS3500-1
type striped
mdisk_id
mdisk_name
fast_write_state empty
used_capacity 1.00GB
real_capacity 1.00GB
free_capacity 0.00MB
overallocation 100
autoexpand
warning
grainsize
se_copy no
easy_tier on
easy_tier_status inactive
tier generic_ssd
tier_capacity 0.00MB
tier generic_hdd
```

```
tier_capacity
1.00GB
```

In Example 8-8 on page 326, we add the volume copy mirror by using the **addvdiskcopy** command.

Example 8-8 addvdiskcopy

```
IBM_2145:ITSO_SVC1:admin>addvdiskcopy -mdiskgrp STGPool_DS5000-1 -vtype striped -unit gb
Volume_no_mirror
Vdisk [23] copy [1] successfully created
```

I/O Time-out Configuration: Optionally, we can configure the I/O Time-out Configuration for mirrored volumes. See more details in 8.4.6, "Configuring I/O Time-out for Mirrored Volumes" on page 331

In Example 8-9, we add the volume copy mirror by using the **addvdiskcopy** command and set the I/O Time-out Configuration for redundancy.

Example 8-9 addvdiskcopy and I/O time-out configuration

```
IBM_2145:ITSO_SVC1:admin>addvdiskcopy -mdiskgrp STGPool_DS5000-1 -vtype striped
-mirrorwritepriority redundancy -unit gb Volume_no_mirror
Vdisk [23] copy [1] successfully created
```

During the synchronization process, you can see the status by using the **lsvdisksyncprogress** command. As shown in Example 8-10, the first time that the status is checked, the synchronization progress is at 48%; note there is an ETA field. The second time that the command is run, the progress status is at 100%, and the synchronization is complete.

Example 8-10 Synchronization

```
IBM_2145:ITSO_SVC1:admin>lsvdisksyncprogress
vdisk_id vdisk_name       copy_id progress estimated_completion_time
23       Volume_no_mirror 1       48       120507203918
IBM_2145:ITSO_SVC1:admin>lsvdisksyncprogress
vdisk_id vdisk_name       copy_id progress estimated_completion_time
23       Volume_no_mirror 1       100
```

Example 8-11 shows the new mirrored volume copy (copy_id 1) has been added and can be seen by using the **lsvdisk** command.

Example 8-11 lsvdisk

```
IBM_2145:ITSO_SVC1:admin>lsvdisk 23
id 23
name Volume_no_mirror
IO_group_id 0
IO_group_name io_grp0
status online
mdisk_grp_id many
mdisk_grp_name many
capacity 1.00GB
type many
formatted no
mdisk_id many
mdisk_name many
```

```
FC_id
FC_name
RC_id
RC_name
vdisk_UID 6005076801AF813F1000000000000019
throttling 0
preferred_node_id 1
fast_write_state empty
cache readwrite
udid
fc_map_count 0
sync_rate 50
copy_count 2
se_copy_count 0
filesystem
mirror_write_priority redundancy

copy_id 0
status online
sync yes
primary yes
mdisk_grp_id 0
mdisk_grp_name STGPool_DS3500-1
type striped
mdisk_id
mdisk_name
fast_write_state empty
used_capacity 1.00GB
real_capacity 1.00GB
free_capacity 0.00MB
overallocation 100
autoexpand
warning
grainsize
se_copy no
easy_tier on
easy_tier_status inactive
tier generic_ssd
tier_capacity 0.00MB
tier generic_hdd
tier_capacity 1.00GB

copy_id 1
status online
sync yes
primary no
mdisk_grp_id 2
mdisk_grp_name STGPool_DS5000-1
type striped
mdisk_id
mdisk_name
fast_write_state empty
used_capacity 1.00GB
real_capacity 1.00GB
free_capacity 0.00MB
overallocation 100
autoexpand
warning
grainsize
se_copy no
```

```
easy_tier on
easy_tier_status inactive
tier generic_ssd
tier_capacity 0.00MB
tier generic_hdd
tier_capacity
1.00GB
```

When adding a volume copy mirror, you can define a mirror with different parameters to the volume copy. Therefore, you can define a thin-provisioned volume copy for a non-volume copy volume and vice versa, which is one way to migrate a non-thin-provisioned volume to a thin-provisioned volume.

> **Volume copy mirror parameters:** To change the parameters of a volume copy mirror, you must delete the volume copy and redefine it with the new values.

Now, we can change the name of the volume just mirrored from `Volume_no_mirror to Volume_mirrored`, as shown in Example 8-12.

Example 8-12 Volume name changes

```
IBM_2145:ITSO_SVC1:admin>chvdisk -name Volume_mirrored Volume_no_mirror
```

8.4.3 Deleting a mirrored volume

The **rmvdiskcopy** command removes a volume copy from a mirrored volume.

Example 8-13 shows the **rmvdiskcopy** command, which is used to remove a mirrored volume copy.

Example 8-13 Removing a copy from mirrored volume

```
IBM_2145:ITSO_SVC2:superuser>svcinfo lsvdisk volume_source
...
Above and below rows has been removed for brevity
...
copy_id 1
status online
sync no
primary no
mdisk_grp_id 1
mdisk_grp_name Pool_RAID5_02
type striped
mdisk_id
mdisk_name
fast_write_state empty
used_capacity 10.00GB
real_capacity 10.00GB
free_capacity 0.00MB
overallocation 100
autoexpand
warning
grainsize
se_copy no
```

```
easy_tier on
easy_tier_status inactive
tier generic_ssd
tier_capacity 0.00MB
tier generic_hdd
tier_capacity 10.00GB

IBM_2145:ITSO_SVC2:superuser>rmvdiskcopy -copy 1 volume_source
```

8.4.4 Splitting a mirrored volume

The **splitvdiskcopy** command creates a new volume in the specified I/O Group from a copy of the specified volume. If the copy that you are splitting is not synchronized, you must use the **-force** parameter. The command fails if you are attempting to remove the only synchronized copy. To avoid this failure, wait for the copy to synchronize, or split the unsynchronized copy from the volume by using the **-force** parameter. You can run the command when either volume copy is offline.

Example 8-14 shows the **splitvdiskcopy** command, which is used to split a mirrored volume. It creates a new volume, `Volume_new` from `Volume_mirrored`.

Example 8-14 Split volume

```
IBM_2145:ITSO_SVC1:admin>splitvdiskcopy -copy 1 -iogrp 0 -name Volume_new Volume_mirrored
Virtual Disk, id [24], successfully created
```

As you can see in Example 8-15, the new volume named `Volume_new` has been created as an independent volume.

Example 8-15 lsvdisk

```
IBM_2145:ITSO_SVC1:admin>lsvdisk Volume_new
id 24
name Volume_new
IO_group_id 0
IO_group_name io_grp0
status online
mdisk_grp_id 2
mdisk_grp_name STGPool_DS5000-1
capacity 1.00GB
type striped
formatted no
mdisk_id
mdisk_name
FC_id
FC_name
RC_id
RC_name
vdisk_UID 6005076801AF813F100000000000001A
throttling 0
preferred_node_id 2
fast_write_state empty
cache readwrite
udid
fc_map_count 0
sync_rate 50
copy_count 1
se_copy_count 0
```

```
filesystem
mirror_write_priority latency

copy_id 0
status online
sync yes
primary yes
mdisk_grp_id 2
mdisk_grp_name STGPool_DS5000-1
type striped
mdisk_id
mdisk_name
fast_write_state empty
used_capacity 1.00GB
real_capacity 1.00GB
free_capacity 0.00MB
overallocation 100
autoexpand
warning
grainsize
se_copy no
easy_tier on
easy_tier_status inactive
tier generic_ssd
tier_capacity 0.00MB
tier generic_hdd
tier_capacity
1.00GB
```

By issuing the command shown in Example 8-14 on page 329, `Volume_mirrored` will no longer have its mirrored copy and a new volume will be created automatically.

8.4.5 Validating a mirrored volume

The **repairvdiskcopy** command detects and optimally, corrects any volume copies that are not identical.

These parameters can be used when attempting to repair a mirrored volume:

- **-medium**: (Optional) Converts sectors that contain different contents into virtual medium errors on the specified volume. This parameter cannot be used with the **-validate** and **-resync** parameters; you must enter one of the three parameters.
- **-resync**: (Optional) Corrects sectors that contain different contents by copying contents from the primary volume copy to other copies on the specified volume. This parameter cannot be used with the **-medium** and **-validate** parameters; you must enter one of the three parameters.
- **-validate**: (Optional) Reports the first difference found on synchronized online copies of the specified volume, on or after the specified **-startlba** value. This parameter cannot be used with the **-medium** and **-resync** parameters; you must enter one of the three parameters.
- **-startlba lba**: (Optional) Specifies a starting logical block address (LBA) on which to begin the command. The LBA must be specified in hex, with a `0x` prefix.

Attention: Be aware of the following points regarding the **repairvdiskcopy** command:

1. Before you run the **repairvdiskcopy** command, ensure that all volume copies are synchronized.

2. Only one **repairvdiskcopy** command can run on a volume at a time. You must wait for the **repairvdiskcopy** command to complete processing before running the command again.

3. After you start the **repairvdiskcopy** command, you cannot use the command to stop processing.

4. The primary copy of a mirrored volume cannot be changed while the **repairvdiskcopy -resync** command is running.

5. The rate for the **repairvdiskcopy** command is controlled by the synchronization rate of the volume that is being repaired. To suspend the repair process, set the synchronization rate of the volume to 0 using the **chvdisk** command.

8.4.6 Configuring I/O Time-out for Mirrored Volumes

In 6.3.x, when using Mirrored Volumes, you can set the attribute *mirror_write_priority*, which is responsible for I/O Time-out Configuration.

There are two possible values for I/O Time-out Configuration:

1. Latency: short time-out prioritizing low host latency (*default value*). Choosing latency indicates a copy that is slow to respond to a write I/O; it becomes unsynchronized if the other copy successfully writes the data.
2. Redundancy: long time-out prioritizing redundancy. Choosing redundancy indicates that a copy (which is slow to respond to a write I/O) causes the response to be delayed until completion to maintain synchronization.

The **chvdisk** or the **addvdiskcopy** command is used to set I/O Time-out Configuration. In Example 8-16, we use the **chvdisk** command to change I/O Time-out Configuration from *latency* to *redundancy*.

Example 8-16 I/O Time-out Configuration by using chvdisk

```
IBM_2145:ITSO_SVC2:superuser>chvdisk -mirrorwritepriority redundancy
WINPRD_MIRROR_VOL01
IBM_2145:ITSO_SVC2:superuser>svcinfo lsvdisk WINPRD_MIRROR_VOL01
id 2
name WINPRD_MIRROR_VOL01
IO_group_id 0
IO_group_name io_grp0
status online
mdisk_grp_id 3
mdisk_grp_name Pool_EASYTIER
capacity 10.00GB
type striped
formatted no
mdisk_id
mdisk_name
FC_id
```

```
FC_name
RC_id
RC_name
vdisk_UID 6005076801AB00E9080000000000003
throttling 0
preferred_node_id 6
fast_write_state empty
cache readwrite
udid
fc_map_count 0
sync_rate 50
copy_count 1
se_copy_count 0
filesystem
mirror_write_priority redundancy
RC_change no
...
Below rows has been removed for brevity
...
```

In Example 8-17, we use the **addvdiskcopy** command to create a mirrored copy of a volume. Configuring I/O Time-out Configuration for *redundancy* as *latency* is the default configuration if you do not specify any parameter for this purpose.

Example 8-17 I/O Time-out Configuration by using addvdiskcopy

```
IBM_2145:ITSO_SVC2:superuser>addvdiskcopy -mdiskgrp 1 -vtype striped
-mirrorwritepriority redundancy ESXI01_VOL01
Vdisk [6] copy [1] successfully created
IBM_2145:ITSO_SVC2:superuser>lsvdisk ESXI01_VOL01
id 6
name ESXI01_VOL01
IO_group_id 0
IO_group_name io_grp0
status online
mdisk_grp_id many
mdisk_grp_name many
capacity 20.00GB
type many
formatted no
mdisk_id many
mdisk_name many
FC_id
FC_name
RC_id
RC_name
vdisk_UID 6005076801AB00E9080000000000007
throttling 0
preferred_node_id 6
fast_write_state empty
cache readwrite
udid
fc_map_count 0
sync_rate 50
copy_count 2
se_copy_count 0
```

```
filesystem
mirror_write_priority redundancy
RC_change no

copy_id 0
status online
sync yes
primary yes
mdisk_grp_id 0
mdisk_grp_name Pool_RAID5_01
type striped
mdisk_id
mdisk_name
fast_write_state empty
used_capacity 20.00GB
real_capacity 20.00GB
free_capacity 0.00MB
overallocation 100
autoexpand
warning
grainsize
se_copy no
easy_tier on
easy_tier_status inactive
tier generic_ssd
tier_capacity 0.00MB
tier generic_hdd
tier_capacity 20.00GB

copy_id 1
status online
sync no
primary no
mdisk_grp_id 1
mdisk_grp_name Pool_RAID5_02
type striped
mdisk_id
mdisk_name
fast_write_state empty
used_capacity 20.00GB
real_capacity 20.00GB
free_capacity 0.00MB
overallocation 100
autoexpand
warning
grainsize
se_copy no
easy_tier on
easy_tier_status inactive
tier generic_ssd
tier_capacity 0.00MB
tier generic_hdd
tier_capacity 20.00GB
```

Our mirrored volume has now been created.

9

Server integration

This chapter explains how to access copied volumes from storage clients running various operating systems, and how to ensure that the copies generated are intact and can be easily accessed when needed.

From a server perspective, there is no difference between a volume that was copied using FlashCopy and a volume that was copied using Metro Mirror or Global Mirror. The hosts simply see a volume with data; how it was created is irrelevant. The focus here is on accessing volumes, regardless of how they were created.

The following topics are discussed:

- Copied data
- Data on copied volumes
- AIX specifics
- Windows specifics
- Linux specifics
- Other operating systems

9.1 Copied data

When a volume is copied using SAN Volume Controller copy services, a full block-by-block replica is created. *Everything* on the source or primary volume is copied to the target or secondary volume. On many systems, not only application data is copied. Many systems write configuration data to the disks to enable volume managers to uniquely identify them and determine their place in volume groups.

This approach can be a benefit or a disadvantage. When presenting the copied volumes to a brand-new server, it is much simpler to bring the volumes back online. However, when presenting these volumes to the same server that accesses the original volumes, there are challenges involved. Because the metadata on the volumes is expected to be unique, presenting the copied volumes breaches that requirement.

Be aware that you cannot selectively copy data from one volume to another; all data is copied, block by block. This chapter discusses techniques for handling this requirement.

9.2 Data on copied volumes

As discussed in Chapter 2, “Planning for Replication Family Services” on page 5, the data on the original volumes might not match what your applications and file systems expect because of the cache between that layer and the SAN Volume Controller (SVC) layer.

It is important to ensure that either:

- Data in the caches above the SVC are fully flushed before initiating the copy.
- Or, the systems that access the copied volumes can recover using the data even though the data is not an identical copy of the system’s perspective.

You can achieve the second option by using journaled file systems and applications that protect their actions with a transaction log.

A critical moment in a copy services mapping relationship’s lifecycle affects the contents of the target or secondary volume. After this critical moment, the readable contents of the volume are no longer updated. This critical moment depends on which copy service is in use:

- For FlashCopy, the critical moment is when the mapping is started.
- For Metro/Global Mirror, the critical moment is when the relationship is stopped (either manually or due to errors).

In this chapter, the action is referred to as *freezing*. The critical moment is referred to as the *freeze point*.

At the freeze point, the target or secondary volume is an image of the source or primary volume at that moment in time (plus or minus a few seconds for Global Mirror). Any data in the cache above SVC will *not* be on the target or secondary volume.

In the case of manual actions, it is relatively straightforward to ensure that the target or secondary volume is a useful copy. For Metro/Global Mirror relationships that are stopped due to SAN errors, it is not possible to ensure that the target or secondary volume is a useful copy, so it is important that the systems accessing these volumes are configured to allow recovery.

Be aware that certain information is *not* copied; for example, persistent reserves are made against a specific logical unit and are not copied with FlashCopy or Metro/Global Mirror.

9.3 AIX specifics

This section describes the steps to create and access copied volumes on AIX hosts.

SVC supports the use of IBM Subsystem Device Driver (SDD), IBM Subsystem Device Driver Path Control Module (SDDPCM), AIX MPIO and third-party drivers on AIX systems to provide multipathing. For details refer to:

http://www-03.ibm.com/systems/storage/software/virtualization/svc/interop.html

When SDD is in use, AIX takes hdisks and creates vpaths. When SDDPCM or AIX MPIO is in use, AIX creates hdisks, and each disk has multiple paths. The commands and outputs are different for different multipath drivers so refer to IBM or third-party documentation for more information about those topics.

For the sake of clarity, the following section assumes the use of SDD and thus discusses vpaths. The methods here apply just as well with SDDPCM and AIX MPIO. However, you need to replace the term *vpath* with the term *hdisk* in the following discussion if you use SDDPCM or AIX MPIO.

9.3.1 JFS, JFS2, and Copy Services

Prior to the freeze point of a remote copy disk or the creation of a new FlashCopy, you must ensure that any information in the file system cache is written to all of the relevant volumes. The simplest way to do so is to unmount the file system, as in Example 9-1.

Example 9-1 Unmount a file system prior to making the break

```
#umount <source filesystem>
```

If unmounting file systems is not an option, an alternative is to force an update of the node table and a flush of buffered files. The AIX freeze and thaw functionality forces an update of the node table and a flush of buffered files. It is available only on Journaled File System 2 (JFS2) file systems. Freezing a file system writes all dirty file system metadata and user data to the disk. While frozen, the file system is read-only, and anything that attempts to modify the file system or its contents must wait for the freeze to end. At this point, the break can be made, and the resulting copy can be easily picked up by another server. Example 9-2 shows how this is done.

Example 9-2 A simple freeze and thaw process

```
#chfs -a freeze=60 <source filesystem>

<freeze the volume(s)>

#chfs -a freeze=0 <source filesystem>
```

In Example 9-2, the file system will be frozen for 60 seconds. This value can be set to whatever time is necessary to complete your Copy Services operations. If you use FlashCopy, you can reduce the amount of time that the freeze must run by starting and completing a prepare operation prior to issuing the JFS2 freeze. The freeze=0 will execute a manual release of the freeze if you want to do so prior to the expiration of the initial freeze timer.

Your setting for the freeze timer needs to be tested with your particular application environment. Although JFS2 might be more than willing to indefinitely delay your I/O, your applications can have their own timeouts.

Sometimes the freeze point occurs at an unplanned time, such as during a data center outage. So long as the mappings are consistent, your file systems must be recoverable, and in the case of a JFS or JFS2 file system, this recovery must not result in any loss to file system metadata. However, application data can still be lost or corrupted if a part of it has still not been sent to the SVC at the time of the outage. After the file data has been received by the SVC, the mirrored, battery-backed cache will preserve the data. Ensuring that the data sent to the SVC is in a consistent state at all times is the responsibility of the application through the use of techniques such as database logs.

9.3.2 Example of copy creation

The following example illustrates the process explained in 9.3.1, “JFS, JFS2, and Copy Services” on page 337. The example performs the following tasks:

- Create new volumes for the FlashCopy targets.
- Create a Consistency Group and put the mappings in it.
- Prepare the Consistency Group.
- Start the Consistency Group.

As shown in Example 9-3, we first retrieve the initial configuration of the SVC cluster presenting volumes to this AIX server. The two volumes represent a data drive and a log drive. Note that the data here is the vdisk_UID, which is a long field near the end.

Example 9-3 Cluster configuration prior to FlashCopy of two volumes

```
IBM_2145:ITSOCL2:admin>svcinfo lsvdisk -delim :
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
5:kanagaFS:0:io_grp0:online:1:DS4700:36.0GB:striped:::::6005076801AD80E8E000000000
000005:0
6:kanagaLog:0:io_grp0:online:0:DS4500:512.0MB:striped:::::6005076801AD80E8E0000000
00000006:0

IBM_2145:ITSOCL2:admin>svcinfo lshost KANAGA
id 2
name KANAGA
port_count 2
type generic
mask 1111
iogrp_count 4
WWPN 10000000C932A7FB
node_logged_in_count 2
state active
WWPN 10000000C932A800
node_logged_in_count 0
state active
```

Example 9-4 shows how these volumes have been configured on an AIX server to hold a JFS2 file system. The vdisk_UID property of a volume matches the SERIAL property of a vpath. Thus, you can use these two properties to determine which vpath represents which volume.

Example 9-4 AIX configuration prior to FlashCopying two volumes

```
#datapath query device

Total Devices : 2


DEV#:   0  DEVICE NAME: vpath0  TYPE: 2145           POLICY:    Optimized
SERIAL: 6005076801AD80E8E000000000000005
==========================================================================
Path#      Adapter/Hard Disk          State     Mode     Select     Errors
    0          fscsi0/hdisk3           OPEN   NORMAL        209          0
    1          fscsi0/hdisk5           OPEN   NORMAL        220          0
    2          fscsi0/hdisk7           OPEN   NORMAL          0          0
    3          fscsi0/hdisk9           OPEN   NORMAL          0          0

DEV#:   1  DEVICE NAME: vpath1  TYPE: 2145           POLICY:    Optimized
SERIAL: 6005076801AD80E8E000000000000006
==========================================================================
Path#      Adapter/Hard Disk          State     Mode     Select     Errors
    0          fscsi0/hdisk4           OPEN   NORMAL          0          0
    1          fscsi0/hdisk6           OPEN   NORMAL          0          0
    2          fscsi0/hdisk8           OPEN   NORMAL         74          0
    3         fscsi0/hdisk10           OPEN   NORMAL         79          0

#lspv
hdisk0          0009cddaea97bf61                    rootvg          active
hdisk1          0009cdda43c9dfd5                    rootvg          active
hdisk2          0009cddabaef1d99                    rootvg          active
hdisk3          none                                None
hdisk4          none                                None
hdisk5          none                                None
hdisk6          none                                None
hdisk7          none                                None
hdisk8          none                                None
hdisk9          none                                None
hdisk10         none                                None
vpath0          0009cdda1b5058e4                    st7s99          active
vpath1          0009cdda1b55c48c                    st7s99          active

#lsvg -l st7s99
st7s99:
LV NAME             TYPE       LPs    PPs    PVs  LV STATE      MOUNT POINT
itsoLV              jfs2       32     32     1    open/syncd    /itsoFS
itsoLog             jfs2log    4      4      1    open/syncd    N/A

#lsfs
Name            Nodename   Mount Pt                 VFS   Size    Options    Auto
Accounting
/dev/hd4        --         /                        jfs2  196608  --         yes  no
/dev/hd1        --         /home                    jfs2  65536   --         yes  no
```

```
/dev/hd2        --        /usr                jfs2  19005440 --        yes  no
/dev/hd9var     --        /var                jfs2  65536   --         yes  no
/dev/hd3        --        /tmp                jfs2  393216  --         yes  no
/proc           --        /proc               procfs --      --        yes  no
/dev/hd10opt    --        /opt                jfs2  2031616 --         yes  no
/dev/itsoLV     --        /itsoFS             jfs2  4194304 rw         no   no

#mount
  node      mounted        mounted over    vfs       date        options
-------- ---------------  ---------------  ------ ------------ ---------------
         /dev/hd4        /                jfs2   Nov 07 09:55 rw,log=/dev/hd8
         /dev/hd2        /usr             jfs2   Nov 07 09:55 rw,log=/dev/hd8
         /dev/hd9var     /var             jfs2   Nov 07 09:55 rw,log=/dev/hd8
         /dev/hd3        /tmp             jfs2   Nov 07 09:55 rw,log=/dev/hd8
         /dev/hd1        /home            jfs2   Nov 07 09:56 rw,log=/dev/hd8
         /proc           /proc            procfs Nov 07 09:56 rw
         /dev/hd10opt    /opt             jfs2   Nov 07 09:56 rw,log=/dev/hd8
         /dev/itsoLV     /itsoFS          jfs2   Nov 07 11:17 rw,log=/dev/itsoLog
```

The Volume Group (VG) that is copied is `st7s99`, which **lspv** tells us is made up of vpaths 0 and 1.

Because the only file system in st7s99 is a JFS2 file system, we can freeze the file system prior to starting the FlashCopy. Write requests on the applications that access this file system are delayed until the file system is thawed. With appropriate planning and scripting, this freeze can be kept to a few seconds at most. To help automate the volume group and host operations, refer to these steps:

1. Create the FlashCopy target volumes:

```
IBM_2145:ITSOCL2:admin>svctask mkvdisk -vtype striped -size 36 -unit gb
-mdiskgrp 1 -iogrp 0 -name kanagaFS-FC
Virtual Disk, id [4], successfully created

IBM_2145:ITSOCL2:admin>svctask mkvdisk -vtype striped -size 512 -unit mb
-mdiskgrp 0 -iogrp 0 -name kanagaLog-FC
Virtual Disk, id [3], successfully created
```

2. Create a FlashCopy Consistency Group:

```
IBM_2145:ITSOCL2:admin>svctask mkfcconsistgrp -name kanagaFC
FlashCopy Consistency Group, id [2], successfully created
```

3. Create FlashCopy mappings for the volumes and add them to the new FlashCopy Consistency Group:

```
IBM_2145:ITSOCL2:admin>svctask mkfcmap -source kanagaFS -target kanagaFS-FC
-consistgrp kanagaFC
FlashCopy Mapping, id [0], successfully created

IBM_2145:ITSOCL2:admin>svctask mkfcmap -source kanagaLog -target kanagaLog-FC
-consistgrp kanagaFC
FlashCopy Mapping, id [1], successfully created
```

4. Prepare the FlashCopy Consistency Group:

```
IBM_2145:ITSOCL2:admin>svctask prestartfcconsistgrp kanagaFC
```

5. Wait until the Consistency Group is in the Prepared state:

```
IBM_2145:ITSOCL2:admin>svcinfo lsfcconsistgrp
```

```
id              name             status
2               kanagaFC         preparing
.
.
.
IBM_2145:ITSOCL2:admin>svcinfo lsfcconsistgrp
id              name             status
2               kanagaFC         prepared
```

6. Freeze the file system on the AIX server:

   ```
   #chfs -a freeze=60 /itsoFS
   ```

7. Start the FlashCopy Consistency Group. This is the freeze point:

   ```
   IBM_2145:ITSOCL2:admin>svctask startfcconsistgrp kanagaFC
   ```

8. Thaw the file system on the AIX server:

   ```
   #chfs -a freeze=0 /itsoFS
   ```

At this point the FlashCopy operation has been started; volumes 3 and 4 are exact copies of volumes 5 and 6 at the time of the file system freeze; and the file system has been unfrozen for user operations.

9.3.3 Accessing useful copies in AIX

If the source or primary volume is defined to the AIX logical volume manager (LVM), all of its data structures and identifiers are also copied to the target or secondary volume including the Volume Group Descriptor Area (VGDA), which contains the physical volume identifier (PVID) and volume group identifier (VGID).

For AIX LVM, it is currently not possible to activate a volume group with a physical volume (vpath) containing a VGID and a PVID that are already used in a volume group existing on the same server.

The restriction still applies even if the vpath PVID is cleared and reassigned with the two commands listed in Example 9-5. These commands only create a new PVID in the ODM database; they do not modify the VGDA on the disk to match. Therefore, the VG is not recognized when you attempt an **importvg**.

Example 9-5 Clearing PVIDs in the ODM (does not make copies accessible)

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

Therefore, it is necessary to redefine the volume group information about the FlashCopy target volume using special procedures or the **recreatevg** command. This command alters the PVIDs and VGIDs in all the VGDAs of the FlashCopy target volumes, so that no conflicts occur with existing PVIDs and VGIDs on existing volume groups residing on the source volumes. If you do not redefine the volume group information prior to importing the volume group, the **importvg** command fails.

Accessing target or secondary Volumes from the same AIX host

This section describes a method of accessing the FlashCopy target volumes on a single AIX host while the source volumes are still active on the same server. Use this procedure as a guide. It might not cover all scenarios.

AIX recreatevg command

Copying source volume content using FlashCopy duplicates all of the data structures and identifiers that are used by AIX LVM to the target volume. The duplicate definitions (PVID and VGID) cause conflicts within LVM, but this problem is solved by issuing the AIX command **recreatevg**. The **recreatevg** command is officially available in all levels of AIX that SVC supports.

The **recreatevg** command overcomes the problem of duplicated LVM data structures and identifiers that is caused by a disk duplication process, such as FlashCopy. You use the **recreatevg** command to recreate an AIX volume group (VG) on a set of target volumes that are copied from a set of source volumes belonging to a specific VG. The command allocates new physical volume identifiers (PVIDs) for the member disks and a new VGID to the volume group. The command also provides options for renaming the logical volumes with a prefix that you specify and options for renaming labels to specify different mount points for file systems. To use this command, you must have root user authority.

> **Important:** Running **recreatevg** changes the PVID on Volumes. In the case of a Metro/Global Mirror relationship, if you decide to reverse the direction of the relationship, you will change the PVID of the original primary volume during the synchronization process. Before mounting the primary volume again, you must modify the PVID again, if you choose to leave the secondary mounted.

The following steps assume that the AIX server and the SVC cluster are still in the same state that they were in at the end of the process that we just described.

Perform these tasks to access the FlashCopy target Volumes:

1. Map the FlashCopy target volumes to the host:

   ```
   IBM_2145:ITSOCL2:admin>svctask mkvdiskhostmap -host KANAGA kanagaFS-FC
   Virtual Disk to Host map, id [2], successfully created
   IBM_2145:ITSOCL2:admin>svctask mkvdiskhostmap -host KANAGA kanagaLog-FC
   Virtual Disk to Host map, id [3], successfully created
   ```

2. Run Configuration Manager to pick up the new devices on the AIX server:

   ```
   #cfgmgr -vl fscsi0
   ```

   (We have omitted the debug output from this command).

3. Run Configuration Manager to create vpaths with the new hdisks:

   ```
   #cfgmgr -vl dpo
   ```

At this point, the configuration of the AIX server has been changed. Example 9-6 shows the new status.

Example 9-6 AIX configuration after FlashCopying two volumes

```
#datapath query device

Total Devices : 4


DEV#:   0  DEVICE NAME: vpath0  TYPE: 2145           POLICY:    Optimized
SERIAL: 6005076801AD80E8E000000000000005
==========================================================================
Path#      Adapter/Hard Disk          State     Mode     Select     Errors
    0          fscsi0/hdisk3           OPEN   NORMAL        214          0
    1          fscsi0/hdisk5           OPEN   NORMAL        232          0
```

```
    2         fscsi0/hdisk7          OPEN   NORMAL         0          0
    3         fscsi0/hdisk9          OPEN   NORMAL         0          0

DEV#:   1  DEVICE NAME: vpath1  TYPE: 2145           POLICY:    Optimized
SERIAL: 6005076801AD80E8E000000000000006
==========================================================================
Path#      Adapter/Hard Disk          State     Mode     Select     Errors
    0         fscsi0/hdisk4          OPEN   NORMAL         0          0
    1         fscsi0/hdisk6          OPEN   NORMAL         0          0
    2         fscsi0/hdisk8          OPEN   NORMAL        91          0
    3        fscsi0/hdisk10          OPEN   NORMAL       102          0

DEV#:   2  DEVICE NAME: vpath2  TYPE: 2145           POLICY:    Optimized
SERIAL: 6005076801AD80E8E000000000000007
==========================================================================
Path#      Adapter/Hard Disk          State     Mode     Select     Errors
    0        fscsi0/hdisk11         CLOSE   NORMAL         0          0
    1        fscsi0/hdisk13         CLOSE   NORMAL         0          0
    2        fscsi0/hdisk15         CLOSE   NORMAL         0          0
    3        fscsi0/hdisk17         CLOSE   NORMAL         0          0

DEV#:   3  DEVICE NAME: vpath3  TYPE: 2145           POLICY:    Optimized
SERIAL: 6005076801AD80E8E000000000000009
==========================================================================
Path#      Adapter/Hard Disk          State     Mode     Select     Errors
    0        fscsi0/hdisk12         CLOSE   NORMAL         0          0
    1        fscsi0/hdisk14         CLOSE   NORMAL         0          0
    2        fscsi0/hdisk16         CLOSE   NORMAL         0          0
    3        fscsi0/hdisk18         CLOSE   NORMAL         0          0

#lspv
hdisk0          0009cddaea97bf61                    rootvg          active
hdisk1          0009cdda43c9dfd5                    rootvg          active
hdisk2          0009cddabaef1d99                    rootvg          active
hdisk3          none                                None
hdisk4          none                                None
hdisk5          none                                None
hdisk6          none                                None
hdisk7          none                                None
hdisk8          none                                None
hdisk9          none                                None
hdisk10         none                                None
vpath0          0009cdda1b5058e4                    st7s99          active
vpath1          0009cdda1b55c48c                    st7s99          active
hdisk11         0009cdda1b5058e4                    st7s99          active
hdisk12         0009cdda1b55c48c                    st7s99          active
hdisk13         0009cdda1b5058e4                    st7s99          active
hdisk14         0009cdda1b55c48c                    st7s99          active
hdisk15         0009cdda1b5058e4                    st7s99          active
hdisk16         0009cdda1b55c48c                    st7s99          active
hdisk17         0009cdda1b5058e4                    st7s99          active
```

```
hdisk18         0009cdda1b55c48c                    st7s99          active
vpath2          none                                None
vpath3          none
None
```

The new vpaths, `vpath2` and `vpath3`, appear with no PVID. The hdisks that make up vpaths 2 and 3 have the same PVIDs as vpaths 1 and 2. Because the PVIDs of the new hdisks and the original vpaths overlap, any attempt to perform an **importvg** command on the new hdisks is unsuccessful.

Note also that vpath2 and vpath3 are in a Close state, because nothing currently accesses them. The following steps recreate the VG with a new name and new PVIDs and recovers the logical volumes (LVs):

1. Create the new VG and prefix all file system path names with /fc and prefix all AIX LVs with FC:

   ```
   #recreatevg -y st7s99fc -L /fc -Y FC vpath2 vpath3
   st7s99fc
   ```

   > **Important:** The LV prefix *cannot* be “fc,” because “fc” is already defined as a prefix in the PdDv class of the device configuration database and is not a valid LV prefix.

2. Run **fsck** on the new file system. Strictly speaking, it is not necessary to run **fsck**, because the steps taken while creating the copy ensure that the file system is client, but it is a prudent defensive step:

   ```
   #fsck -y /fc/itsoFS:

   The current volume is: /dev/FCitsoLV
   Primary superblock is valid.
   J2_LOGREDO:log redo processing for /dev/FCitsoLV
   Primary superblock is valid.
   *** Phase 1 - Initial inode scan
   *** Phase 2 - Process remaining directories
   *** Phase 3 - Process remaining files
   *** Phase 4 - Check and repair inode allocation map
   *** Phase 5 - Check and repair block allocation map
   File system is clean.
   ```

3. Mount the new file system:

   ```
   #mount /fc/itsoFS
   Replaying log for /dev/FCitsoLV
   ```

At this point, the FlashCopy target volumes are ready for access. Example 9-7 shows the final server state. This new file system contains all of the files that were in the original file system at the time of the freeze.

Example 9-7 AIX configuration after recreating the VG

```
#datapath query device

Total Devices : 4


DEV#:   0  DEVICE NAME: vpath0  TYPE: 2145           POLICY:    Optimized
SERIAL: 6005076801AD80E8E000000000000005
```

```
==========================================================================
Path#      Adapter/Hard Disk          State     Mode     Select     Errors
    0          fscsi0/hdisk3           OPEN   NORMAL        217          0
    1          fscsi0/hdisk5           OPEN   NORMAL        237          0
    2          fscsi0/hdisk7           OPEN   NORMAL          0          0
    3          fscsi0/hdisk9           OPEN   NORMAL          0          0

DEV#:   1  DEVICE NAME: vpath1  TYPE: 2145            POLICY:    Optimized
SERIAL: 6005076801AD80E8E000000000000006
==========================================================================
Path#      Adapter/Hard Disk          State     Mode     Select     Errors
    0          fscsi0/hdisk4           OPEN   NORMAL          0          0
    1          fscsi0/hdisk6           OPEN   NORMAL          0          0
    2          fscsi0/hdisk8           OPEN   NORMAL         98          0
    3         fscsi0/hdisk10           OPEN   NORMAL        109          0

DEV#:   2  DEVICE NAME: vpath2  TYPE: 2145            POLICY:    Optimized
SERIAL: 6005076801AD80E8E000000000000007
==========================================================================
Path#      Adapter/Hard Disk          State     Mode     Select     Errors
    0         fscsi0/hdisk11           OPEN   NORMAL        202          0
    1         fscsi0/hdisk13           OPEN   NORMAL        206          0
    2         fscsi0/hdisk15           OPEN   NORMAL          0          0
    3         fscsi0/hdisk17           OPEN   NORMAL          0          0

DEV#:   3  DEVICE NAME: vpath3  TYPE: 2145            POLICY:    Optimized
SERIAL: 6005076801AD80E8E000000000000009
==========================================================================
Path#      Adapter/Hard Disk          State     Mode     Select     Errors
    0         fscsi0/hdisk12           OPEN   NORMAL          0          0
    1         fscsi0/hdisk14           OPEN   NORMAL          0          0
    2         fscsi0/hdisk16           OPEN   NORMAL         61          0
    3         fscsi0/hdisk18           OPEN   NORMAL         57          0

#lspv
hdisk0          0009cddaea97bf61                    rootvg          active
hdisk1          0009cdda43c9dfd5                    rootvg          active
hdisk2          0009cddabaef1d99                    rootvg          active
hdisk3          none                                None
hdisk4          none                                None
hdisk5          none                                None
hdisk6          none                                None
hdisk7          none                                None
hdisk8          none                                None
hdisk9          none                                None
hdisk10         none                                None
vpath0          0009cdda1b5058e4                    st7s99          active
vpath1          0009cdda1b55c48c                    st7s99          active
hdisk11         none                                None
hdisk12         none                                None
hdisk13         none                                None
hdisk14         none                                None
hdisk15         none                                None
hdisk16         none                                None
hdisk17         none                                None
```

```
hdisk18         none                               None
vpath2          0009cdda1bf46d57                   st7s99fc        active
vpath3          0009cdda1bf46f3b                   st7s99fc        active

#lsvg
rootvg
st7s99
st7s99fc

#lsvg -l st7s99
st7s99:
LV NAME             TYPE       LPs    PPs    PVs  LV STATE      MOUNT POINT
itsoLV              jfs2       32     32     1    open/syncd    /itsoFS
itsoLog             jfs2log    4      4      1    open/syncd    N/A

#lsvg -l st7s99fc
st7299fc:
LV NAME             TYPE       LPs    PPs    PVs  LV STATE      MOUNT POINT
FCitsoLV            jfs2       32     32     1    open/syncd    /fc/itsoFS
FCitsoLog           jfs2log    4      4      1    open/syncd    N/A

#lsfs
Name            Nodename   Mount Pt               VFS    Size     Options    Auto
Accounting
/dev/hd4        --         /                      jfs2   196608   --         yes  no
/dev/hd1        --         /home                  jfs2   65536    --         yes  no
/dev/hd2        --         /usr                   jfs2   19005440 --         yes  no
/dev/hd9var     --         /var                   jfs2   65536    --         yes  no
/dev/hd3        --         /tmp                   jfs2   393216   --         yes  no
/proc           --         /proc                  procfs --        --        yes  no
/dev/hd10opt    --         /opt                   jfs2   2031616 --          yes  no
/dev/itsoLV     --         /itsoFS                jfs2   4194304 rw          no   no
/dev/FCitsoLV   --         /fc/itsoFS             jfs2   4194304 rw          no   no

#mount
node   mounted          mounted over    vfs       date        options
------ ---------------  --------------- ------ ------------ ---------------
       /dev/hd4         /               jfs2   Nov 07 09:55 rw,log=/dev/hd8
       /dev/hd2         /usr            jfs2   Nov 07 09:55 rw,log=/dev/hd8
       /dev/hd9var      /var            jfs2   Nov 07 09:55 rw,log=/dev/hd8
       /dev/hd3         /tmp            jfs2   Nov 07 09:55 rw,log=/dev/hd8
       /dev/hd1         /home           jfs2   Nov 07 09:56 rw,log=/dev/hd8
       /proc            /proc           procfs Nov 07 09:56 rw
       /dev/hd10opt     /opt            jfs2   Nov 07 09:56 rw,log=/dev/hd8
       /dev/itsoLV      /itsoFS         jfs2   Nov 07 11:17 rw,log=/dev/itsoLog
       /dev/FCitsoLV    /fc/itsoFS      jfs2   Nov 07 13:15 rw,log=/dev/FCitsoLog
```

Accessing target or secondary volumes from a different AIX host

Accessing target or secondary volumes from a different server than the server accessing the source or primary volumes is much simpler.

The following procedure makes the data of the FlashCopy target volume available to another AIX host that has no prior definitions of the target volume in its configuration database, which is the Object Data Manager (ODM). Example 9-8 shows the state of the AIX server prior to presenting the target volumes.

Example 9-8 AIX configuration prior to presenting FlashCopy target Volumes

```
#datapath query device

No device file found

#lsvg
rootvg

#lspv
hdisk0          0009cddaea97bf61                    rootvg          active
hdisk1          0009cdda43c9dfd5                    rootvg          active
hdisk2          0009cddabaef1d99                    rootvg          active

#lsfs
Name            Nodename   Mount Pt               VFS   Size    Options    Auto
Accounting
/dev/hd4        --         /                      jfs2  196608  --         yes  no
/dev/hd1        --         /home                  jfs2  65536   --         yes  no
/dev/hd2        --         /usr                   jfs2  19005440 --        yes  no
/dev/hd9var     --         /var                   jfs2  65536   --         yes  no
/dev/hd3        --         /tmp                   jfs2  393216  --         yes  no
/proc           --         /proc                  procfs --      --        yes  no
/dev/hd10opt    --         /opt                   jfs2  2031616 --         yes  no
```

The initial process for accessing the Volumes is the same as before:

1. Map the FlashCopy target volumes to the host:

   ```
   IBM_2145:ITSOCL2:admin>svctask mkvdiskhostmap -host UMNAK kanagaFS-FC
   Virtual Disk to Host map, id [2], successfully created
   IBM_2145:ITSOCL2:admin>svctask mkvdiskhostmap -host UMNAK kanagaLog-FC
   Virtual Disk to Host map, id [3], successfully created
   ```

2. Run Configuration Manager to pick up the new devices on the AIX server:

   ```
   #cfgmgr -vl fscsi0
   ```

   (We skipped the debug output from this command.)

3. Run Configuration Manager to create vpaths with the new hdisks:

   ```
   #cfgmgr -vl dpo
   ```

At this point, the configuration of the AIX server has been changed. Example 9-9 shows the new status.

Example 9-9 AIX configuration after presenting FlashCopy target volumes

```
datapath query device

Total Devices : 2


DEV#:   0  DEVICE NAME: vpath2  TYPE: 2145           POLICY:    Optimized
SERIAL: 6005076801AD80E8E000000000000007
==========================================================================
Path#      Adapter/Hard Disk          State     Mode     Select     Errors
```

```
    0          fscsi0/hdisk3         CLOSE   NORMAL         0          0
    1          fscsi0/hdisk5         CLOSE   NORMAL         0          0
    2          fscsi0/hdisk7         CLOSE   NORMAL         0          0
    3          fscsi0/hdisk9         CLOSE   NORMAL         0          0

DEV#:   1  DEVICE NAME: vpath3  TYPE: 2145           POLICY:    Optimized
SERIAL: 6005076801AD80E8E000000000000009
==========================================================================
Path#      Adapter/Hard Disk         State     Mode     Select     Errors
    0          fscsi0/hdisk4         CLOSE   NORMAL         0          0
    1          fscsi0/hdisk6         CLOSE   NORMAL         0          0
    2          fscsi0/hdisk8         CLOSE   NORMAL         0          0
    3         fscsi0/hdisk10         CLOSE   NORMAL         0          0

#lspv
hdisk0         0009cddaea97bf61                    rootvg          active
hdisk1         0009cdda43c9dfd5                    rootvg          active
hdisk2         0009cddabaef1d99                    rootvg          active
hdisk3         none                                None
hdisk4         none                                None
hdisk5         none                                None
hdisk6         none                                None
hdisk7         none                                None
hdisk8         none                                None
hdisk9         none                                None
hdisk10        none                                None
vpath2         0009cdda1b5058e4                    st7s99
vpath3         0009cdda1b55c48c                    st7s99
```

At this time, note that the new vpaths have PVIDs, and the hdisks have none. We use the **importvg** command to import the information about the VG into this server's ODM:

1. Import the target volume group:

   ```
   #importvg -y st7s99 vpath2
   st7s99
   ```

2. Vary on the Volume Group (the **importvg** command varies on the volume group):

   ```
   #varyonvg st7s99
   ```

3. Verify consistency of all file systems on the FlashCopy target volume:

   ```
   #fsck -y /itsoFS


   The current volume is: /dev/itsoLV
   Primary superblock is valid.
   J2_LOGREDO:log redo processing for /dev/itsoLV
   Primary superblock is valid.
   *** Phase 1 - Initial inode scan
   *** Phase 2 - Process remaining directories
   *** Phase 3 - Process remaining files
   *** Phase 4 - Check and repair inode allocation map
   *** Phase 5 - Check and repair block allocation map
   File system is clean.
   ```

4. Mount all the target file systems:

   `#mount /itsoFS`

At this point, the FlashCopy target volumes are ready for access. Example 9-10 shows the final server state. This new file system contains all of the files that were in the original file system at the time of the freeze.

Example 9-10 AIX configuration after importing the VG

```
#datapath query device

Total Devices : 2


DEV#:   0  DEVICE NAME: vpath2  TYPE: 2145           POLICY:    Optimized
SERIAL: 6005076801AD80E8E000000000000007
==========================================================================
Path#      Adapter/Hard Disk          State     Mode     Select     Errors
    0          fscsi0/hdisk3           OPEN   NORMAL        150          0
    1          fscsi0/hdisk5           OPEN   NORMAL        161          0
    2          fscsi0/hdisk7           OPEN   NORMAL          0          0
    3          fscsi0/hdisk9           OPEN   NORMAL          0          0

DEV#:   1  DEVICE NAME: vpath3  TYPE: 2145           POLICY:    Optimized
SERIAL: 6005076801AD80E8E000000000000009
==========================================================================
Path#      Adapter/Hard Disk          State     Mode     Select     Errors
    0          fscsi0/hdisk4           OPEN   NORMAL          0          0
    1          fscsi0/hdisk6           OPEN   NORMAL          0          0
    2          fscsi0/hdisk8           OPEN   NORMAL         40          0
    3         fscsi0/hdisk10           OPEN   NORMAL         44          0

#lspv
hdisk0         0009cddaea97bf61                    rootvg         active
hdisk1         0009cdda43c9dfd5                    rootvg         active
hdisk2         0009cddabaef1d99                    rootvg         active
hdisk3         none                                None
hdisk4         none                                None
hdisk5         none                                None
hdisk6         none                                None
hdisk7         none                                None
hdisk8         none                                None
hdisk9         none                                None
hdisk10        none                                None
vpath2         0009cdda1b5058e4                    st7s99         active
vpath3         0009cdda1b55c48c                    st7s99         active

#lsvg
rootvg
st7s99

#lsvg -l st7s99
st7s99:
LV NAME             TYPE       LPs    PPs    PVs  LV STATE      MOUNT POINT
itsoLV              jfs2       32     32     1    open/syncd    /itsoFS
itsoLog             jfs2log    4      4      1    open/syncd    N/A
```

```
#lsfs
Name            Nodename   Mount Pt               VFS   Size    Options    Auto
Accounting
/dev/hd4        --         /                      jfs2  196608  --         yes  no
/dev/hd1        --         /home                  jfs2  65536   --         yes  no
/dev/hd2        --         /usr                   jfs2  19005440 --        yes  no
/dev/hd9var     --         /var                   jfs2  65536   --         yes  no
/dev/hd3        --         /tmp                   jfs2  393216  --         yes  no
/proc           --         /proc                  procfs --      --        yes  no
/dev/hd10opt    --         /opt                   jfs2  2031616 --         yes  no
/dev/itsoLV     --         /itsoFS                jfs2  4194304 rw         no   no

#mount
  node       mounted        mounted over    vfs       date        options
-------- ---------------  ---------------  ------ ------------ ---------------
         /dev/hd4         /                jfs2   Nov 07 14:04 rw,log=/dev/hd8
         /dev/hd2         /usr             jfs2   Nov 07 14:04 rw,log=/dev/hd8
         /dev/hd9var      /var             jfs2   Nov 07 14:04 rw,log=/dev/hd8
         /dev/hd3         /tmp             jfs2   Nov 07 14:04 rw,log=/dev/hd8
         /dev/hd1         /home            jfs2   Nov 07 14:05 rw,log=/dev/hd8
         /proc            /proc            procfs Nov 07 14:05 rw
         /dev/hd10opt     /opt             jfs2   Nov 07 14:05 rw,log=/dev/hd8
         /dev/itsoLV      /itsoFS          jfs2   Nov 07 14:36 rw,log=/dev/itsoLog
```

9.4 Windows specifics

This section provides examples for Microsoft Windows server and IBM SVC Replication Family integration. Because Microsoft Windows Server 2003 and Microsoft Windows Server 2008 have differences in the interface and Microsoft Windows Server 2003 is still a popular and widely used Microsoft Server platform, we show two examples: Microsoft Windows 2000 Server and Windows Server 2003 with IBM SDD driver and Microsoft Windows Server 2008 with Microsoft native driver.

9.4.1 Windows 2000 Server, Windows Server 2003, SVC Advanced Copy Services

Windows 2000 Server and Windows Server 2003 incorporate a simplified version of the VERITAS Volume Manager, which is called the Logical Disk Manager (LDM), for handling their disks.

With the LDM, you can create logical partitions, perform disk mounts, and create dynamic volumes. The five types of dynamic volumes are simple, spanned, mirrored, striped, and RAID-5.

On earlier versions of Windows, the information relating to the disks was stored in the Windows registry. With Windows 2000 Server and Windows Server 2003, this information is stored on the disk drive in a partition called the LDM database, which is kept on the last few tracks of the disk. Each volume has its own LDM database and belongs to a disk group, which is similar to the concept of a physical volume identifier (PVID) and a volume group in AIX. Because the LDM is stored on the physical drive, with Windows 2000 Server and Windows Server 2003, it is possible to move disk drives between different computers.

LDM database consideration: FlashCopy source and target volumes cannot be mapped to the same Windows host when they are created as dynamic volumes, because each dynamic volume *must* have its own unique LDM database. When you perform FlashCopy, the LDM database is also copied, which means that if you tried to mount the source and target volume on the same host system, you have two volumes with exactly the same LDM database. Having two volumes with exactly the same LDM database is not allowed, and you will not be able to mount the target volume.

There is a Microsoft utility, **`dmpss`**, that is capable of modifying the LDM database, but its use for this particular purpose is not supported by either IBM or Microsoft, so using it in this way is at your own risk.

Creating useful copies with Windows

As with AIX, you create useful copies by ensuring that no data is in the system write cache when you freeze the volumes. The methods for ensuring that no data is in the system write cache when you freeze the Volumes are conceptually similar for AIX and Windows. All of these operations require Administrator access on your Windows host.

One option is to remove the drive letter associated with the disk. Figure 9-1 on page 352 shows a volume that is currently being used by a Windows server.

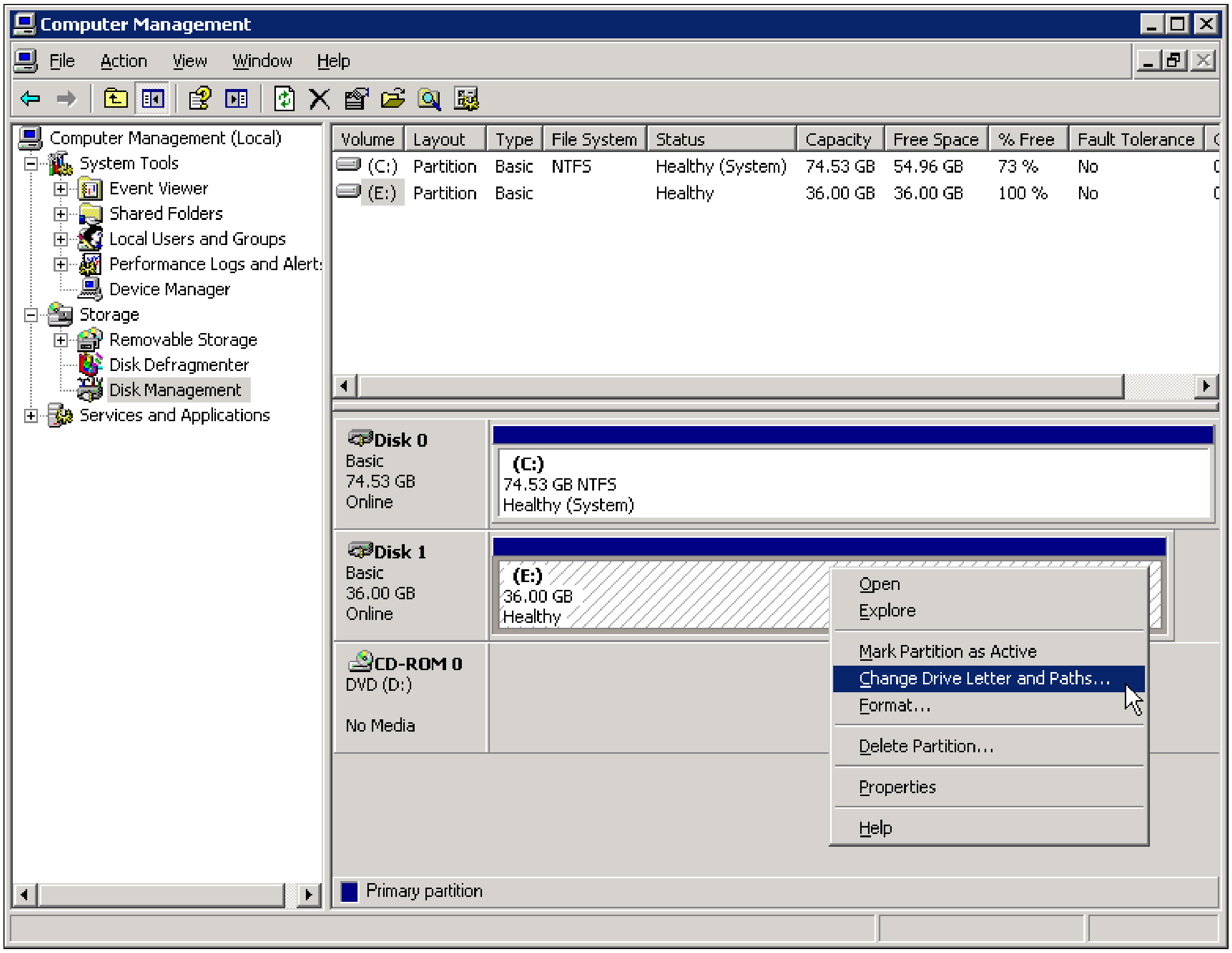


Figure 9-1 Change the drive letter of a Windows disk

Selecting **Change Drive Letter and Paths** opens the dialog box shown in Figure 9-2 on page 352. Removing a drive letter from a Windows disk results in completing all outstanding writes to the disk.

Like unmounting a file system in AIX, no applications can access data on this drive without the drive letter.

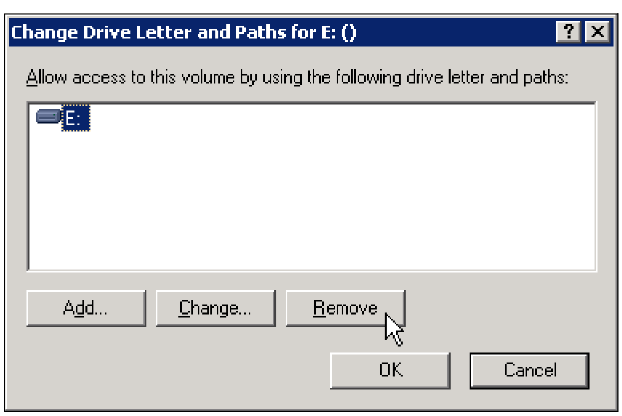


Figure 9-2 Remove a drive letter from a Windows disk

An alternative method is to disable the write cache for the disk in question. Figure 9-3 on page 353 shows using the Computer Management console to access the properties of a Windows disk.

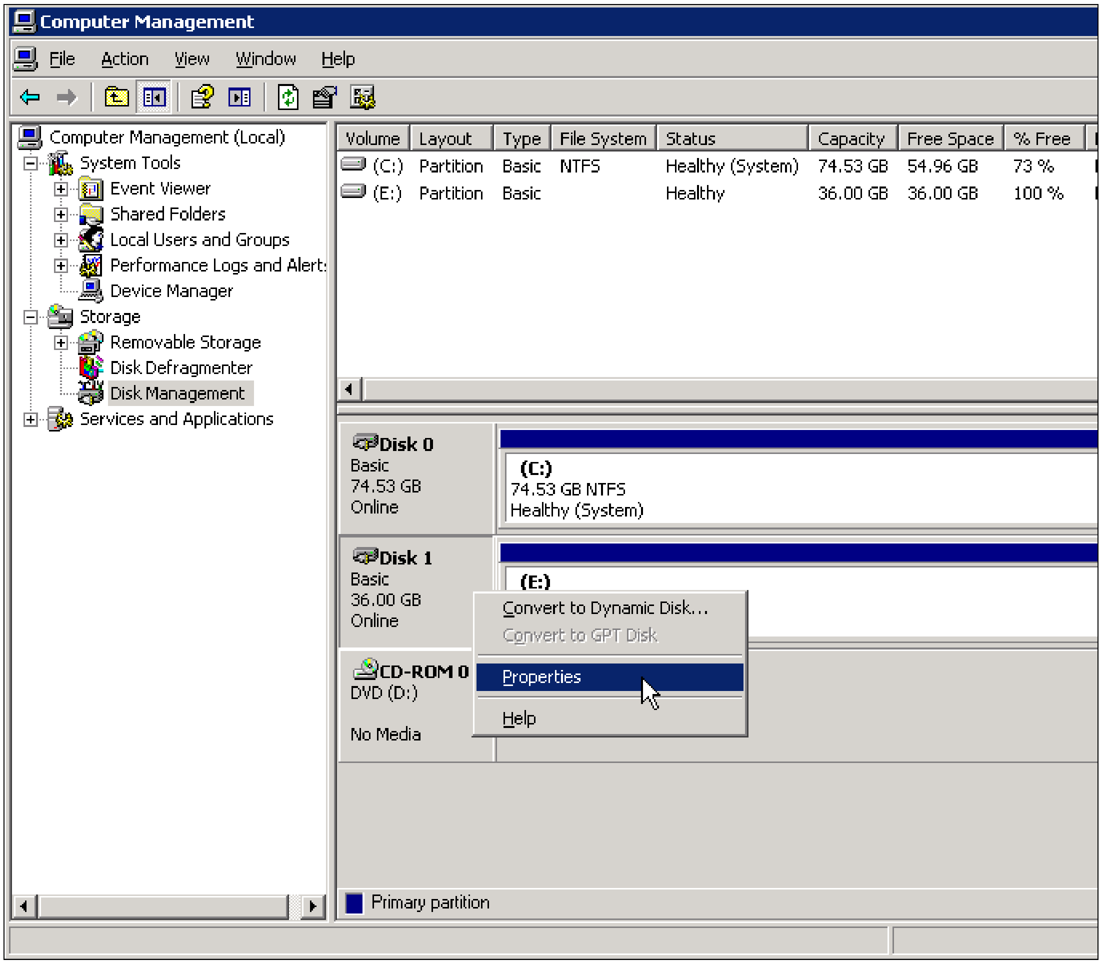


Figure 9-3 Set the properties of a Windows disk

Choosing **Properties** on the window that is shown in Figure 9-3 brings up the dialog box that is shown in Figure 9-4 on page 354.

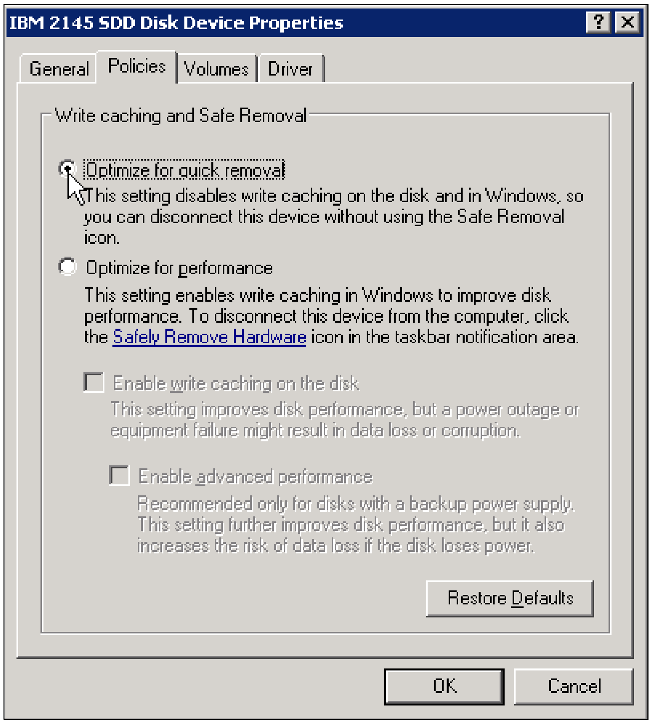


Figure 9-4 Disable the write cache of a Windows disk

By selecting **Optimize for quick removal**, the write cache is disabled, which enables you to make the break and be sure that the volume contents match the image on the host.

> **Consistency consideration:** Although disabling the write cache ensures that the data on the host will be consistent for a particular point in time, it does *not* ensure that the data is actually intact. For instance, if you start the FlashCopy mapping when the host is halfway through writing a 1 GB file, your FlashCopy will certainly miss the 500 MB of data that has not yet been written. If using this method to ensure data integrity, you must make sure that your applications are also in a consistent state.

Example 9-11 shows the initial configuration of the SVC cluster presenting volumes to this Windows server. The two volumes represent a data drive and a log drive.

Example 9-11 Cluster configuration prior to FlashCopying two volumes

```
IBM_2145:ITSOCL2:admin>svcinfo lsvdisk -delim :
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
4:senegalDynamic2:0:io_grp0:online:1:DS4700:36.0GB:striped:2:fcmap2:::6005076801AD
80E8E00000000000000C:1
5:senegalDynamic1:0:io_grp0:online:1:DS4700:36.0GB:striped:1:fcmap1:::6005076801AD
80E8E00000000000000B:1
6:senegalBasic:0:io_grp0:online:1:DS4700:36.0GB:striped:0:fcmap0:::6005076801AD80E
8E00000000000000A:1

IBM_2145:ITSOCL2:admin>svcinfo lshost SENEGAL
id 1
```

```
name SENEGAL
port_count 2
type generic
mask 1111
iogrp_count 4
WWPN 210000E08B89B9C0
node_logged_in_count 2
state active
WWPN 210000E08B89CCC2
node_logged_in_count 0
state active
```

Example 9-12 shows the Subsystem Device Driver (SDD) configuration of the Windows Server 2003 server to which these volumes are mapped. Figure 9-5 on page 356 shows how these volumes have been configured on a Windows Server 2003 server.

Example 9-12 SDD configuration of Windows Server 2003 server

```
C:\Program Files\IBM\Subsystem Device Driver>datapath query device

Total Devices : 3


DEV#:   0  DEVICE NAME: Disk1 Part0  TYPE: 2145       POLICY: OPTIMIZED
SERIAL: 6005076801AD80E8E00000000000000C
============================================================================
Path#            Adapter/Hard Disk        State  Mode       Select     Errors
    0     Scsi Port3 Bus0/Disk1 Part0     OPEN   NORMAL          8          0
    1     Scsi Port3 Bus0/Disk1 Part0     OPEN   NORMAL         10          0
    2     Scsi Port3 Bus0/Disk1 Part0     OPEN   NORMAL          0          0
    3     Scsi Port3 Bus0/Disk1 Part0     OPEN   NORMAL          0          0
    4     Scsi Port3 Bus0/Disk1 Part0     OPEN   NORMAL          0          0
    5     Scsi Port3 Bus0/Disk1 Part0     OPEN   NORMAL          0          0

DEV#:   1  DEVICE NAME: Disk2 Part0  TYPE: 2145       POLICY: OPTIMIZED
SERIAL: 6005076801AD80E8E00000000000000B
============================================================================
Path#            Adapter/Hard Disk        State  Mode       Select     Errors
    0     Scsi Port3 Bus0/Disk2 Part0     OPEN   NORMAL          0          0
    1     Scsi Port3 Bus0/Disk2 Part0     OPEN   NORMAL          0          0
    2     Scsi Port3 Bus0/Disk2 Part0     OPEN   NORMAL          3          0
    3     Scsi Port3 Bus0/Disk2 Part0     OPEN   NORMAL          5          0
    4     Scsi Port3 Bus0/Disk2 Part0     OPEN   NORMAL          5          0
    5     Scsi Port3 Bus0/Disk2 Part0     OPEN   NORMAL          5          0

DEV#:   2  DEVICE NAME: Disk3 Part0  TYPE: 2145       POLICY: OPTIMIZED
SERIAL: 6005076801AD80E8E00000000000000A
============================================================================
Path#            Adapter/Hard Disk        State  Mode       Select     Errors
    0     Scsi Port3 Bus0/Disk3 Part0     OPEN   NORMAL         11          0
    1     Scsi Port3 Bus0/Disk3 Part0     OPEN   NORMAL          9          0
    2     Scsi Port3 Bus0/Disk3 Part0     OPEN   NORMAL          0          0
    3     Scsi Port3 Bus0/Disk3 Part0     OPEN   NORMAL          0          0
    4     Scsi Port3 Bus0/Disk3 Part0     OPEN   NORMAL          0          0
    5     Scsi Port3 Bus0/Disk3 Part0     OPEN   NORMAL          0          0
```

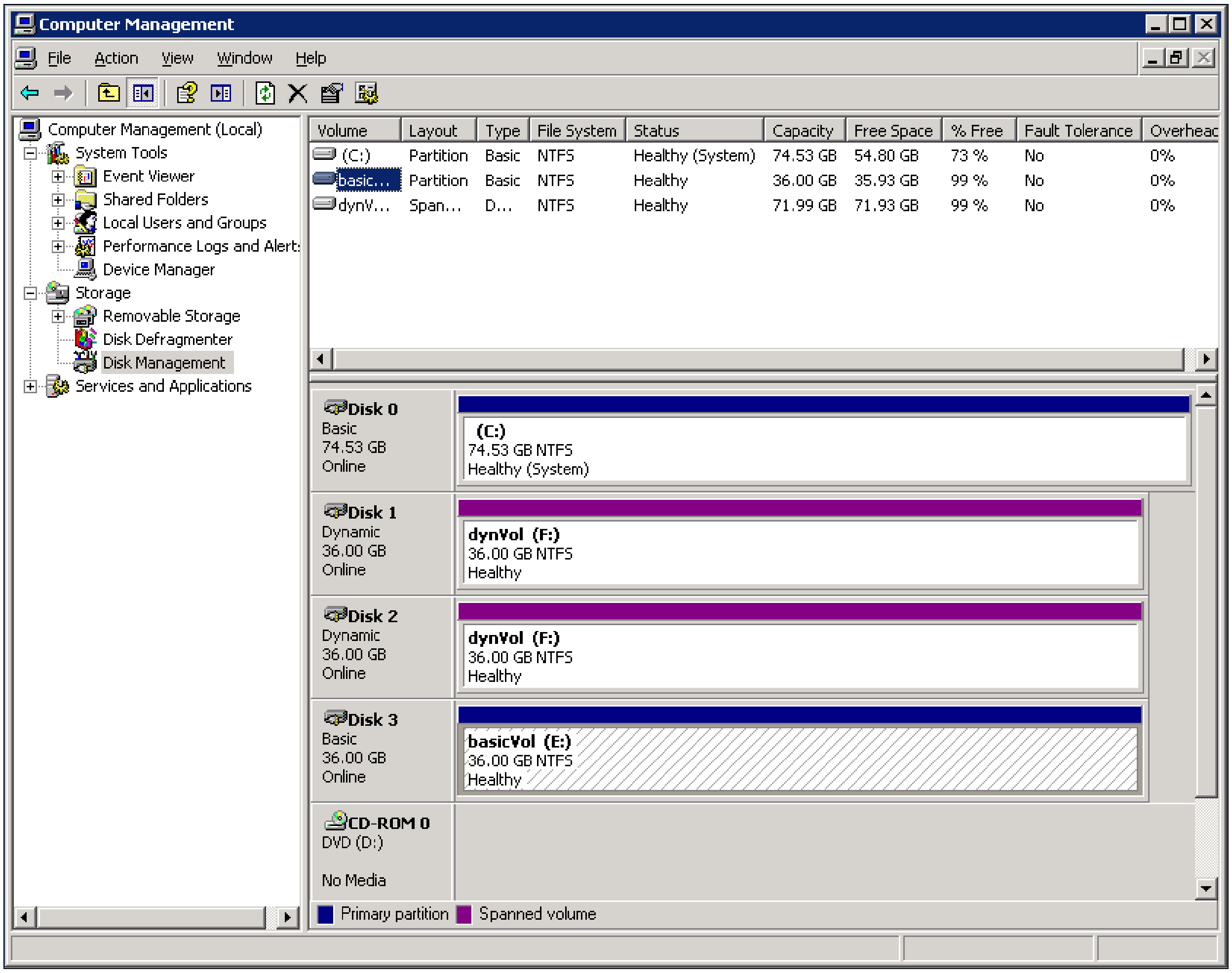


Figure 9-5 Volumes appearing on a Windows Server 2003 host prior to mapping target or secondary Volumes

The vdisk_UID property of a volume matches the SERIAL property of a vpath. The device name of the vpath matches the disk property that is shown under the Volumes tab (see

Figure 9-6 on page 357). Thus, you can use these properties to determine which Windows disk represents which vpath. Subsequently, you can match a Windows disk to a volume.

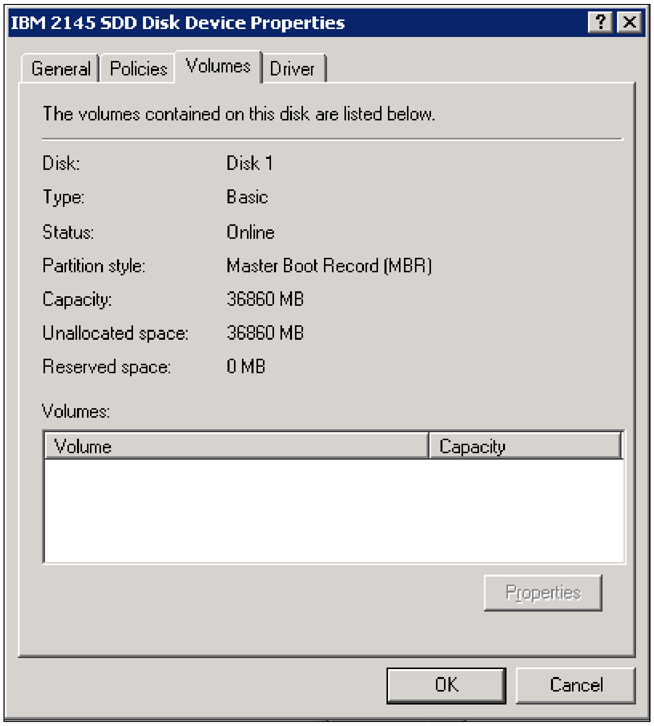


Figure 9-6 Identifying a Windows disk as a volume

The following process shows how you can make sure that the FlashCopy target volumes contain useful copies:

1. Create the FlashCopy target volumes:

```
IBM_2145:ITSOCL2:admin>svctask mkvdisk -vtype striped -unit gb -size 36 -iogrp
0 -mdiskgrp DS4700 -name senegalBasic-FC
Virtual Disk, id [3], successfully created
IBM_2145:ITSOCL2:admin>svctask mkvdisk -vtype striped -unit gb -size 36 -iogrp
0 -mdiskgrp DS4700 -name senegalDyn1-FC
Virtual Disk, id [8], successfully created
IBM_2145:ITSOCL2:admin>svctask mkvdisk -vtype striped -unit gb -size 36 -iogrp
0 -mdiskgrp DS4700 -name senegalDyn2-FC
Virtual Disk, id [7], successfully created
```

2. Create a FlashCopy Consistency Group:

```
IBM_2145:ITSOCL2:admin>svctask mkfcconsistgrp -name senegalFC
FlashCopy Consistency Group, id [1], successfully created
```

3. Create FlashCopy mappings for the volumes and add them to the new FlashCopy Consistency Group:

```
IBM_2145:ITSOCL2:admin>svctask mkfcmap -source senegalBasic -target
senegalBasic-FC -consistgrp senegalFC
FlashCopy Mapping, id [0], successfully created
IBM_2145:ITSOCL2:admin>svctask mkfcmap -source senegalDynamic1 -target
senegalDyn1-FC -consistgrp senegalFC
FlashCopy Mapping, id [1], successfully created
```

```
IBM_2145:ITSOCL2:admin>svctask mkfcmap -source senegalDynamic2 -target
senegalDyn2-FC -consistgrp senegalFC
FlashCopy Mapping, id [2], successfully created
```

4. Prepare the FlashCopy Consistency Group:

```
IBM_2145:ITSOCL2:admin>svctask prestartfcconsistgrp senegalFC
```

5. Wait until the Consistency Group is in the Prepared state:

```
IBM_2145:ITSOCL2:admin>svcinfo lsfcconsistgrp
id              name              status
1               senegalFC         preparing
.
.
.
IBM_2145:ITSOCL2:admin>svcinfo lsfcconsistgrp
id              name              status
1               senegalFC         prepared
```

6. Disable the write cache for the Windows disks, shown in Figure 9-4 on page 354.
7. Start the FlashCopy Consistency Group. This point is the freeze point:

```
IBM_2145:ITSOCL2:admin>svctask startfcconsistgrp senegalFC
```

8. Re-enable the write cache for the Windows disks.

At this point, the FlashCopy operation has been started, and VDisks 3, 8, and 7 are exact copies of VDisks 6, 5, and 4 when the FlashCopy Consistency Group is started.

Unmounting and remounting volumes with the diskpart utility

To aid in scripting, the operation of unmounting and remounting disks can be done with the windows command-line utility **diskpart**.

In our example, we use a diskpart script to unmount a set of drives from a Windows server. The server on which the script runs has the disk configuration that is shown in the Disk Manager window, Figure 9-7 on page 359.

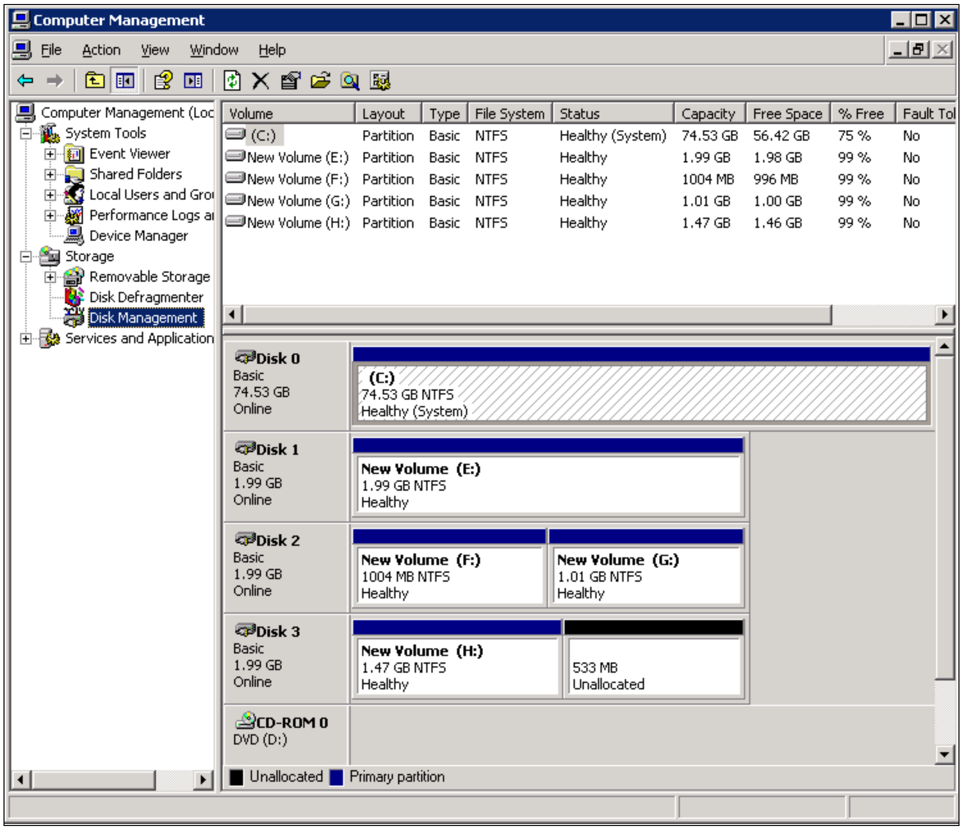


Figure 9-7 Example server disk manager window

In Example 9-13, we use SDD utilities and the **diskpart** utility to view the configuration of the SVC disks from the perspective of the host. As with the GUI example starting in “Accessing useful copies” on page 362, we must cross-reference the Serial number in the **datapath query device** command output with the vdisk_UID in the SVC.

Example 9-13 Using SDD utilities

```
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\Program Files\IBM\SDDDSM>datapath query device

Total Devices : 3


DEV#:   0  DEVICE NAME: Disk1 Part0  TYPE: 2145       POLICY: OPTIMIZED
SERIAL: 60050768019900E9100000000000001
============================================================================
Path#            Adapter/Hard Disk         State  Mode        Select     Errors
    0      Scsi Port2 Bus0/Disk1 Part0     OPEN   NORMAL         849          0
    1      Scsi Port3 Bus0/Disk1 Part0     OPEN   NORMAL         712          0
    2      Scsi Port3 Bus0/Disk1 Part0     OPEN   NORMAL           0          0
    3      Scsi Port2 Bus0/Disk1 Part0     OPEN   NORMAL           0          0
```

```
DEV#:   1  DEVICE NAME: Disk2 Part0  TYPE: 2145       POLICY: OPTIMIZED
SERIAL: 60050768019900E9100000000000002
============================================================================
Path#           Adapter/Hard Disk       State  Mode       Select     Errors
    0     Scsi Port2 Bus0/Disk2 Part0     OPEN   NORMAL          0          0
    1     Scsi Port3 Bus0/Disk2 Part0     OPEN   NORMAL          0          0
    2     Scsi Port3 Bus0/Disk2 Part0     OPEN   NORMAL       1776          0
    3     Scsi Port2 Bus0/Disk2 Part0     OPEN   NORMAL       1429          0

DEV#:   2  DEVICE NAME: Disk3 Part0  TYPE: 2145       POLICY: OPTIMIZED
SERIAL: 60050768019900E9100000000000000
============================================================================
Path#           Adapter/Hard Disk       State  Mode       Select     Errors
    0     Scsi Port3 Bus0/Disk3 Part0     OPEN   NORMAL         73          0
    1     Scsi Port2 Bus0/Disk3 Part0     OPEN   NORMAL          0          0
    2     Scsi Port2 Bus0/Disk3 Part0     OPEN   NORMAL       4132          0
    3     Scsi Port3 Bus0/Disk3 Part0     OPEN   NORMAL       4015          0

C:\Program Files\IBM\SDDDSM>diskpart

Microsoft DiskPart version 5.2.3790.3959
Copyright (C) 1999-2001 Microsoft Corporation.
On computer: SENEGAL

DISKPART> list disk

  Disk ###  Status      Size     Free     Dyn  Gpt
  --------  ----------  -------  -------  ---  ---
  Disk 0    Online        75 GB      0 B
  Disk 1    Online      2039 MB      0 B
  Disk 2    Online      2039 MB      0 B
  Disk 3    Online      2039 MB   533 MB

DISKPART> list volume

  Volume ###  Ltr  Label        Fs     Type        Size     Status     Info
  ----------  ---  -----------  -----  ----------  -------  ---------  --------
  Volume 0     D                       DVD-ROM         0 B  Healthy
  Volume 1     F   New Volume   NTFS   Partition   1004 MB  Healthy
  Volume 2     G   New Volume   NTFS   Partition   1035 MB  Healthy
  Volume 3     E   New Volume   NTFS   Partition   2039 MB  Healthy
  Volume 4     C                NTFS   Partition     75 GB  Healthy    System
  Volume 5     H   New Volume   NTFS   Partition   1506 MB  Healthy

DISKPART> select disk 1

Disk 1 is now the selected disk.

DISKPART> list partition

  Partition ###  Type              Size     Offset
  -------------  ----------------  -------  -------
  Partition 1    Primary           2039 MB    32 KB

DISKPART> select disk 2
```

```
Disk 2 is now the selected disk.

DISKPART> list partition

  Partition ###  Type              Size     Offset
  -------------  ----------------  -------  -------
  Partition 1    Primary           1004 MB    32 KB
  Partition 2    Primary           1035 MB  1004 MB

DISKPART> select disk 3

Disk 3 is now the selected disk.

DISKPART> list partition

  Partition ###  Type              Size     Offset
  -------------  ----------------  -------  -------
  Partition 1    Primary           1506 MB    32 KB

DISKPART> exit

Leaving DiskPart...

C:\Program Files\IBM\SDDDSM>
```

The **diskpart** utility has the ability to run scripts. Unfortunately, building those scripts is something of a challenge. The **diskpart** utility requires the user to select either a disk and partition, or the volume. The problem is that the number assigned to the disks and volumes can change with every reboot; any scripts that use **diskpart** to unmount drives must be dynamically generated based on the output of the **list volume** operation, which lists the volume number associated with each drive letter. Alternatively, you can use the serial number from the SDD command **datapath query device** to sort out the disks.

Precise ways of dynamically building this script are beyond the intended scope of this book. However, we demonstrate what the script looks like in the end for the example in Example 9-13 on page 359. Our example removes the mount points for the previous configuration in drive letter order. The **noerr** option allows the remainder of the script to complete if one operation results in an error, such as a file being open on a drive that you were attempting to dismount. Even with the **noerr** option, **diskpart** as a whole continues to execute the script.

Example 9-14 Sample diskpart umount script

```
select volume 3
remove letter=e: noerr
select volume 1
remove letter=f: noerr
select volume 2
remove letter=g: noerr
select volume 5
remove letter=h: noerr
```

After our FlashCopy operation, we remount the drives. To remount those same drives, we run the script in Example 9-15 on page 362.

Example 9-15 Remount script

```
select volume 3
assign letter=e: noerr
select volume 1
assign letter=f: noerr
select volume 2
assign letter=g: noerr
select volume 5
assign letter=h: noerr
```

Accessing useful copies

Basic disks are relatively straightforward to access from a Windows 2000 Server or a Windows Server 2003 host. However, dynamic disks are a little more complicated because of the restrictions that are imposed on duplicate LDM databases.

> **Dynamic disk consideration:** Target or auxiliary volumes that are dynamic disks must *never* be presented to a host that can see the source or primary volume, because you risk losing data.

Accessing target or auxiliary basic Volumes from the same Windows host

In this section we describe a method of accessing the FlashCopy target volumes on a single Windows host while the source Volumes are still active on the same server.

Perform these tasks to access the FlashCopy target volumes:

1. Map the FlashCopy target volumes to the host:

```
IBM_2145:ITSOCL2:admin>svctask mkvdiskhostmap -host SENEGAL senegalBasic1-FC
Virtual Disk to Host map, id [3], successfully created
IBM_2145:ITSOCL2:admin>svctask mkvdiskhostmap -host SENEGAL senegalBasic2-FC
Virtual Disk to Host map, id [4], successfully created
IBM_2145:ITSOCL2:admin>svctask mkvdiskhostmap -host SENEGAL senegalBasic3-FC
Virtual Disk to Host map, id [5], successfully created
```

2. Scan for new hardware changes, as shown in Figure 9-8 on page 363, to pick up the newly presented volumes.

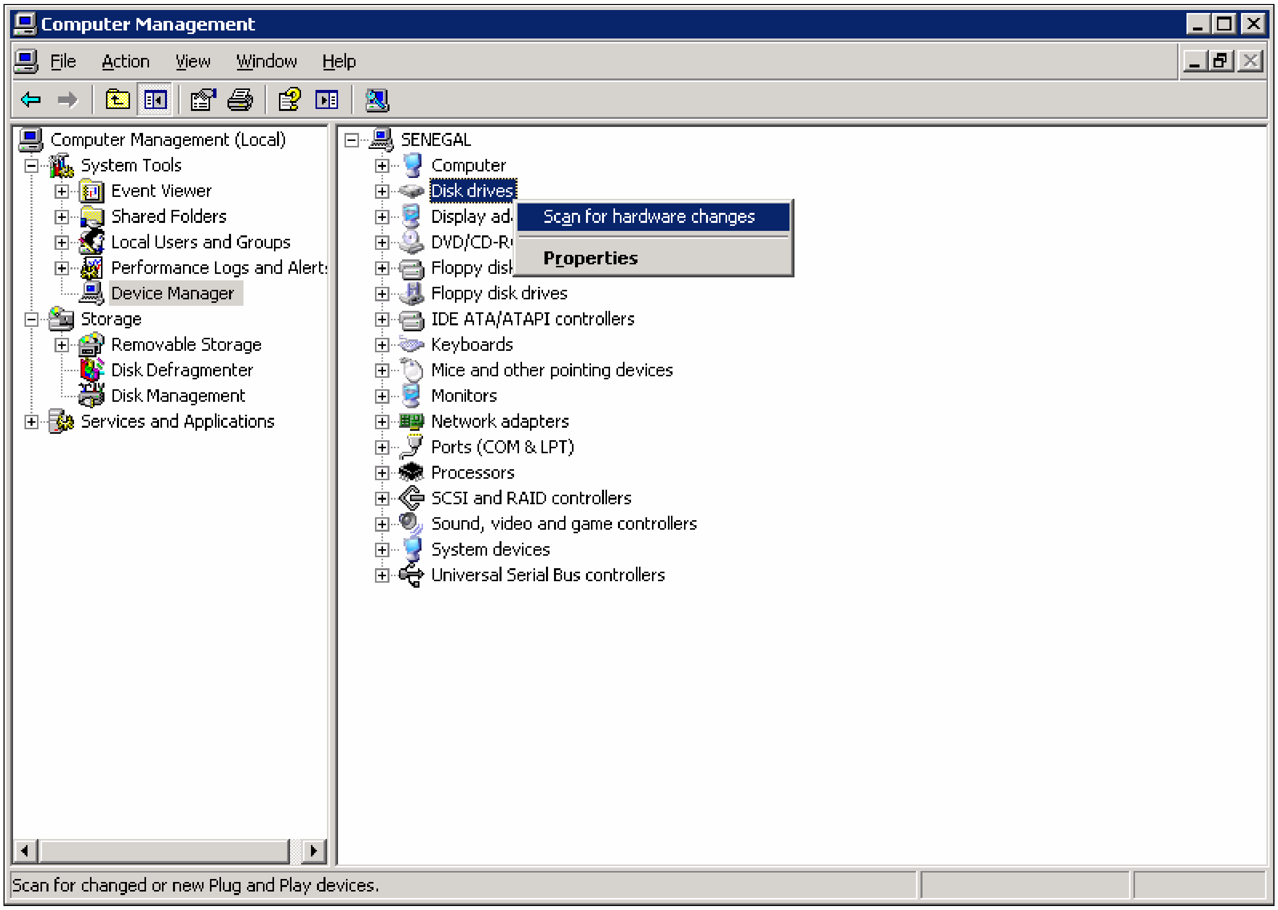


Figure 9-8 Scan for hardware changes to pick up target or auxiliary volumes

At this point, the configuration of the Windows host has changed. Figure 9-9 on page 364 shows the new status. The new disks, Disks 4, 5, and 6 are the FlashCopy disks. Note that the order of the new disks is completely different from the order of the old disks. For this reason, any scripts that you might write to assign drive letters will have to use the **datapath**

query device output to sort out which disk is which. To use the GUI, you only need to assign a drive letter to the drive as shown previously in Figure 9-1 on page 352.

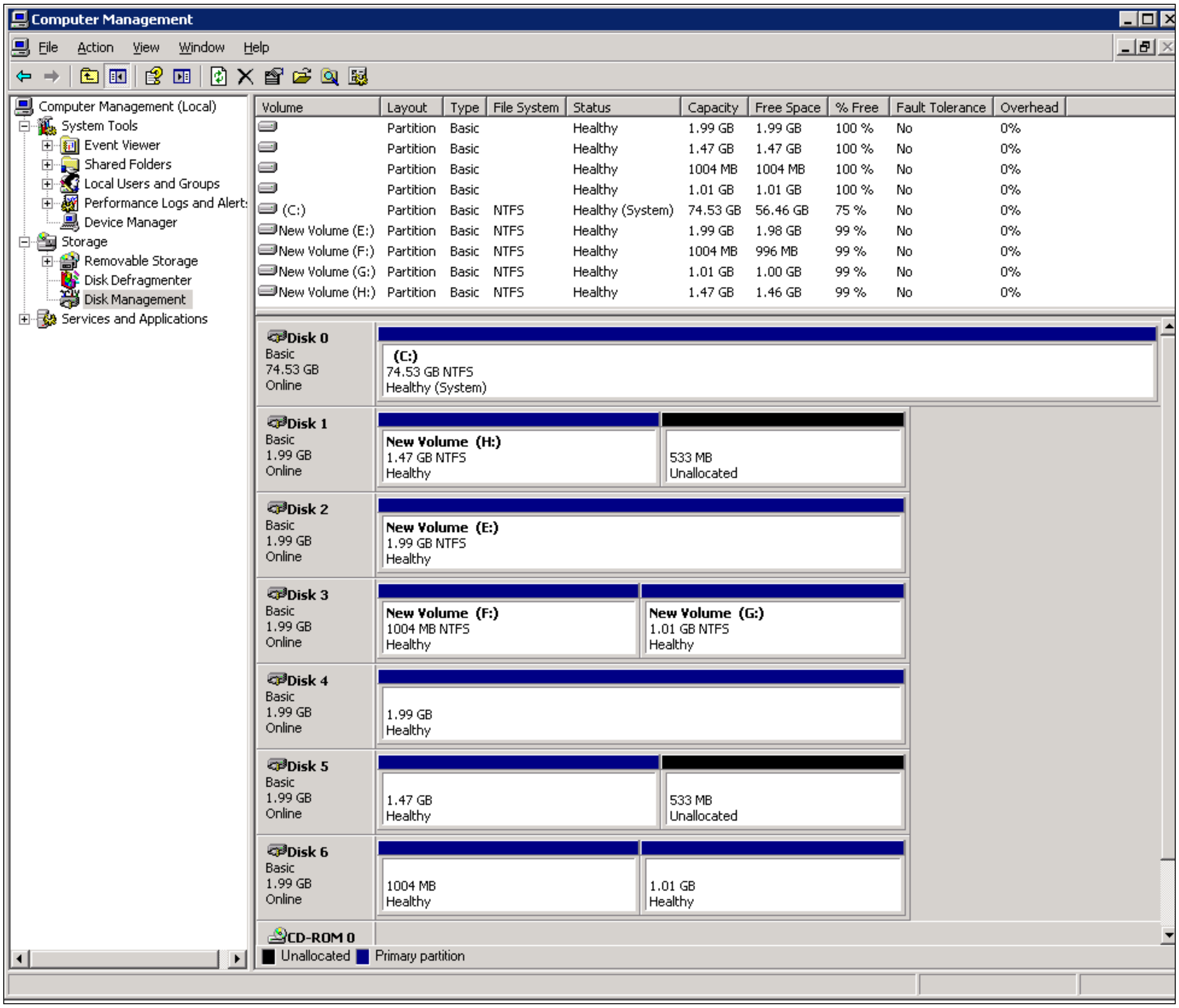


Figure 9-9 Windows configuration after discovering target or secondary VDisks

After clicking **Add** (see Figure 9-2 on page 352), we are presented with the dialog box that is shown in Figure 9-10.

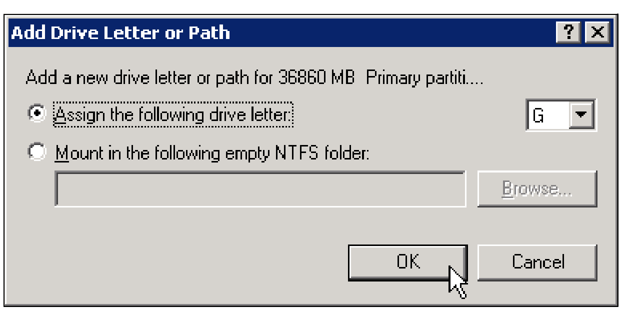


Figure 9-10 Add a drive letter to a basic drive

When you have assigned drive letters to your basic drives, you must run **chkdsk** to ensure file system consistency as shown in Example 9-16 on page 365. This task is particularly important if the break is due to an unplanned Metro/Global Mirror stoppage.

Example 9-16 Run chkdsk on a newly mapped target or secondary volume

```
C:\Program Files\IBM\Subsystem Device Driver>chkdsk /f e:
The type of the file system is NTFS.
Volume label is basicVol.

CHKDSK is verifying files (stage 1 of 3)...
File verification completed.
CHKDSK is verifying indexes (stage 2 of 3)...
Index verification completed.
CHKDSK is verifying security descriptors (stage 3 of 3)...
Security descriptor verification completed.
Windows has checked the file system and found no problems.

  37744685 KB total disk space.
        20 KB in 2 files.
         4 KB in 9 indexes.
         0 KB in bad sectors.
     67137 KB in use by the system.
     65536 KB occupied by the log file.
  37677524 KB available on disk.

      4096 bytes in each allocation unit.
   9436171 total allocation units on disk.
   9419381 allocation units available on disk.
```

Accessing target or secondary volumes from a different Windows host

Accessing target or secondary volumes from a different Windows server than the Windows server that is accessing the source or primary Volumes is much simpler. In Example 9-17, the server originally had no volumes mapped to it, but now we can see that the volumes are mapped to the server.

Example 9-17 Map FlashCopy targets to a new Windows server

```
IBM_2145:ITSOCL2:admin>svctask mkvdiskhostmap -host SIAM senegalBasic-FC
Virtual Disk to Host map, id [2], successfully created
IBM_2145:ITSOCL2:admin>svctask mkvdiskhostmap -host SIAM senegalDyn1-FC
Virtual Disk to Host map, id [3], successfully created
IBM_2145:ITSOCL2:admin>svctask mkvdiskhostmap -host SIAM senegalDyn2-FC
Virtual Disk to Host map, id [4], successfully created
```

As shown in Figure 9-8 on page 363, the next step is to scan for hardware changes. The result of this action is shown in Figure 9-11 on page 366.

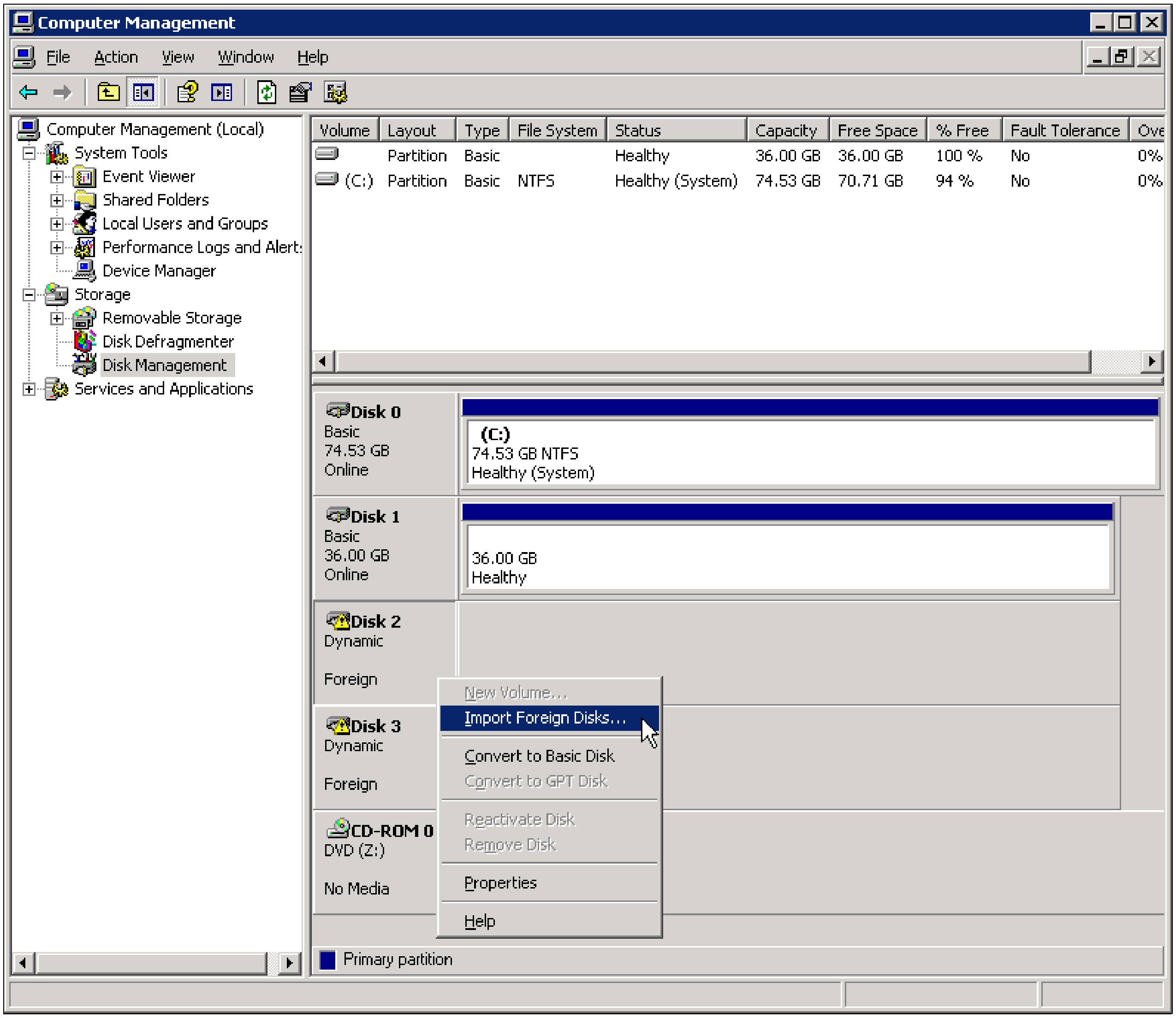


Figure 9-11 Results of hardware scan on a Windows Server 2003 server

As shown, the basic disk is picked up normally and needs a drive letter assigned to it. Figure 9-11 also shows the state of the dynamic disks; they register as foreign.

By selecting **Import Foreign Disks**, you access the dialog box that is shown in Figure 9-12.

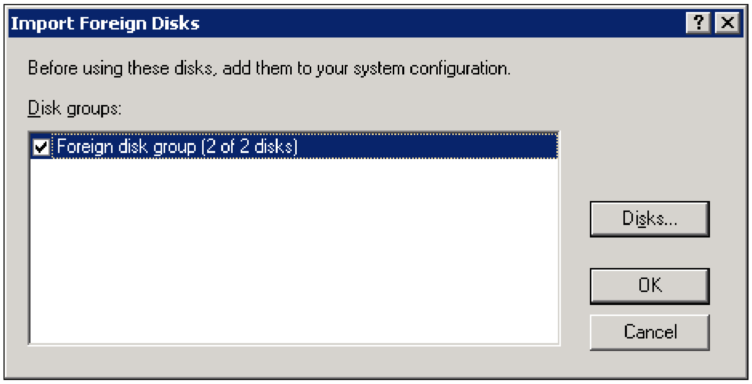


Figure 9-12 Import Foreign Disks

After you click **OK**, you access Figure 9-13, which lists the drives that you will import.

.

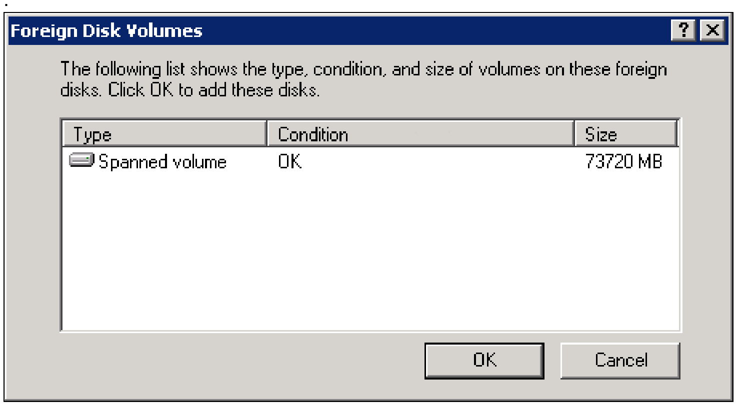


Figure 9-13 Foreign Disk Volumes

At this point, all of the drive letters have been assigned to your volumes; you must run **`chkdsk`** to ensure file system consistency as shown previously in Example 9-16 on page 365. This task is particularly important if the freezing was due to an unplanned Metro/Global Mirror stoppage.

You can also import foreign disks by using the **`diskpart`** utility command **`import`** to bring the disks online. Simply **`select`** one of the disks, and **`import`** will bring them all into Windows.

9.4.2 Microsoft Windows Server 2008 and IBM SVC Advanced Copy Services

It is popular these days to use operating system native drivers. This approach has benefits and disadvantages. General guidance is to use a common driver across the whole environment. In this example we use Windows Server 2008 with its native driver.

Creating Basic Disk copy

We created a new volume on SVC storage and mapped it to the Windows host. It appears automatically in the Disk Management menu as Disk 9 in our case. We change it to `online` and initialize it as shown in Figure 9-14 on page 368.

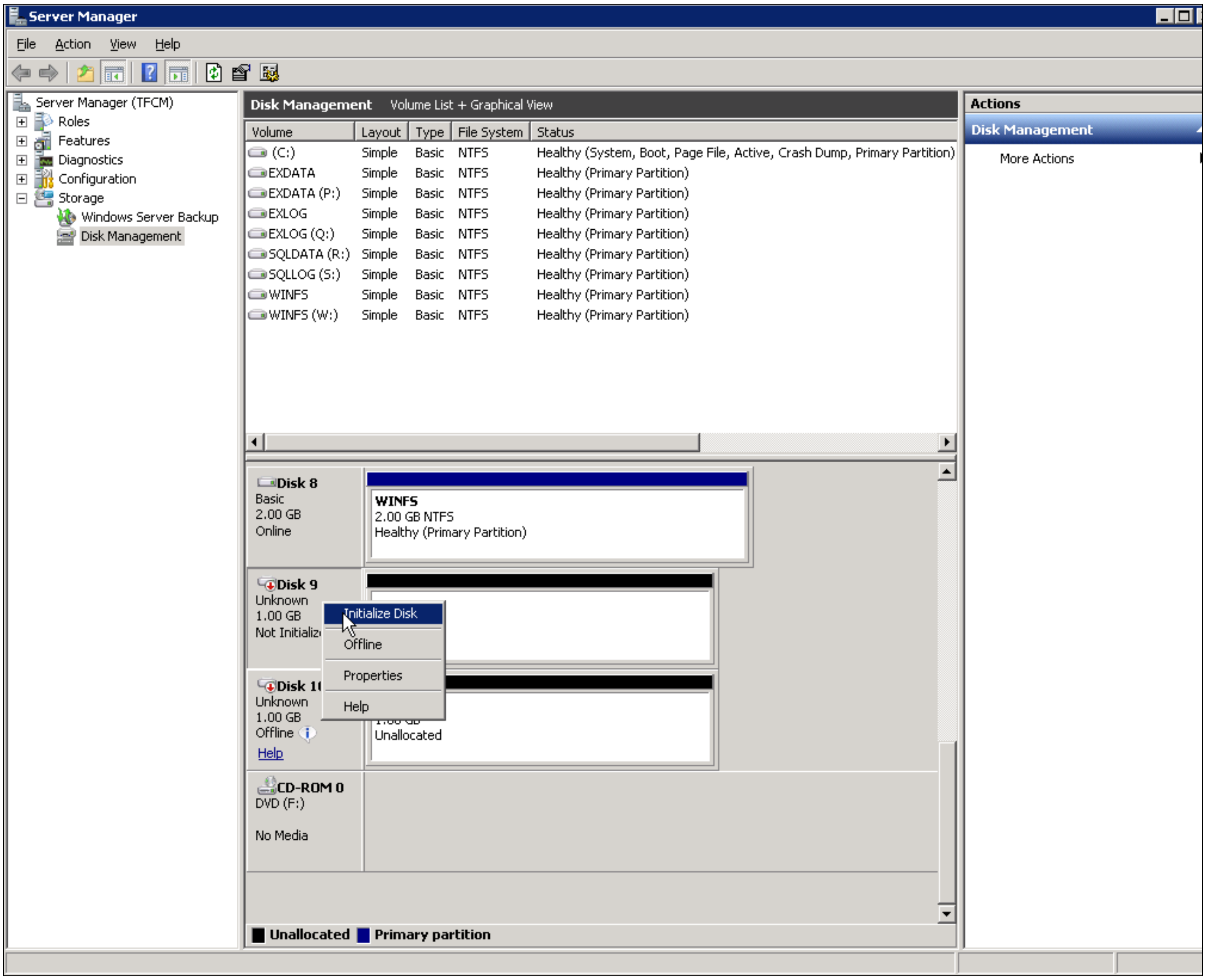


Figure 9-14 Basic disk initialization

Next, we need to choose the partition style: Master Boot Record (MBR) or GUID Partition Table (GPT). GPT is recommended for larger than 2 TB disks or Intel Itanium systems. In both cases the disk is Basic and all FlashCopy operations work right away. We show the process for MBR (Figure 9-15), but exactly the same also applies for GPT partitions.

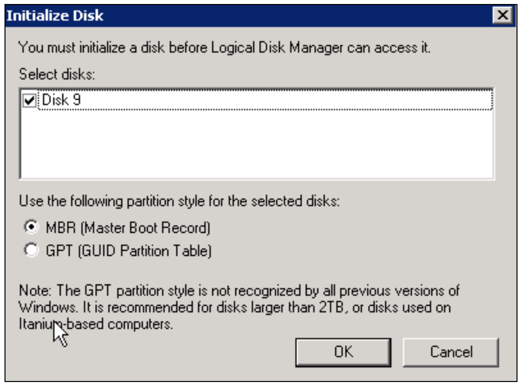


Figure 9-15 Partition style

Then we make a Simple volume with the NTFS file system and mount it as shown in Figure 9-16.

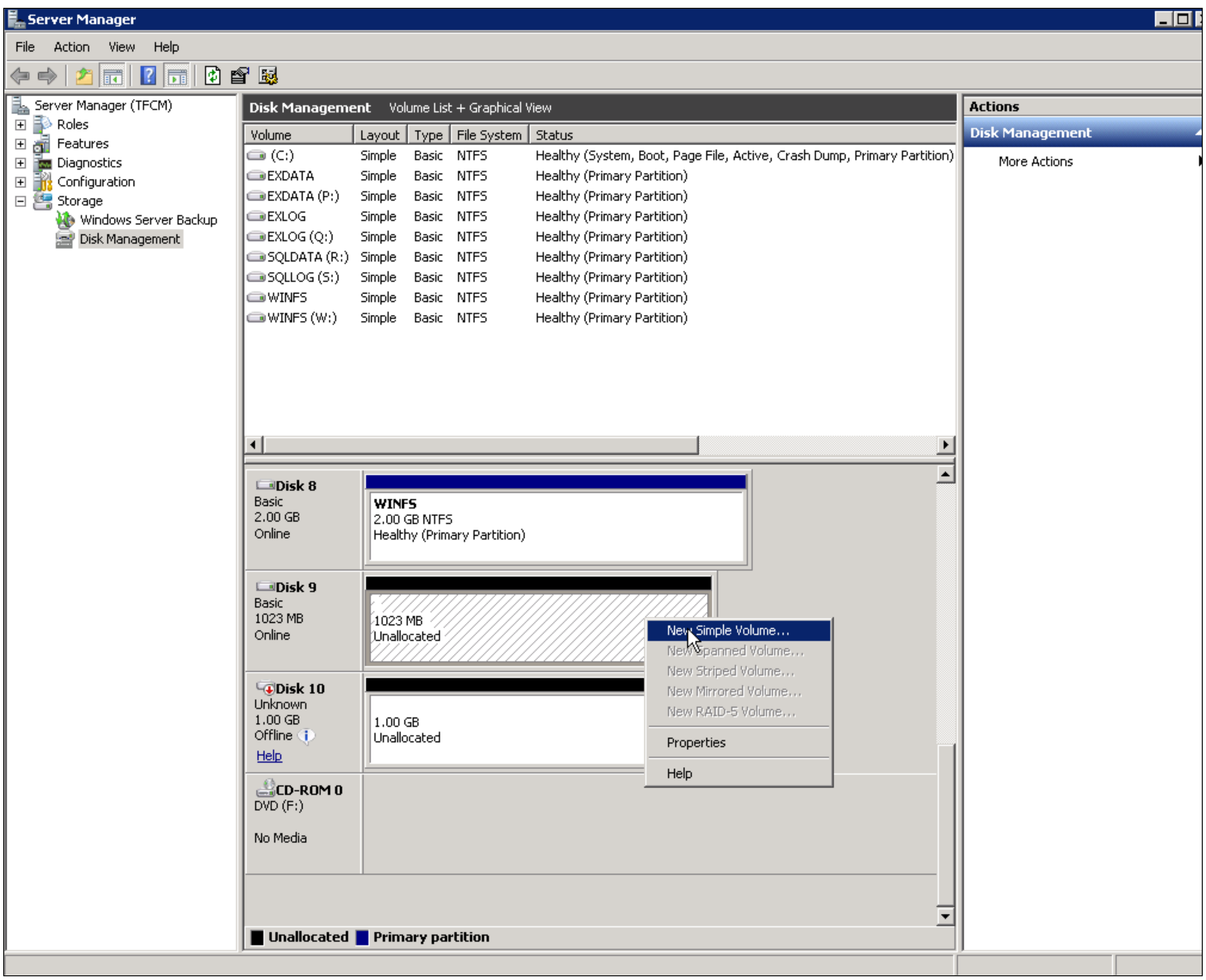


Figure 9-16 Simple volume with NTFS file system

The new disk drive letter is I, and we have copied some files there as shown Figure 9-17 on page 370.

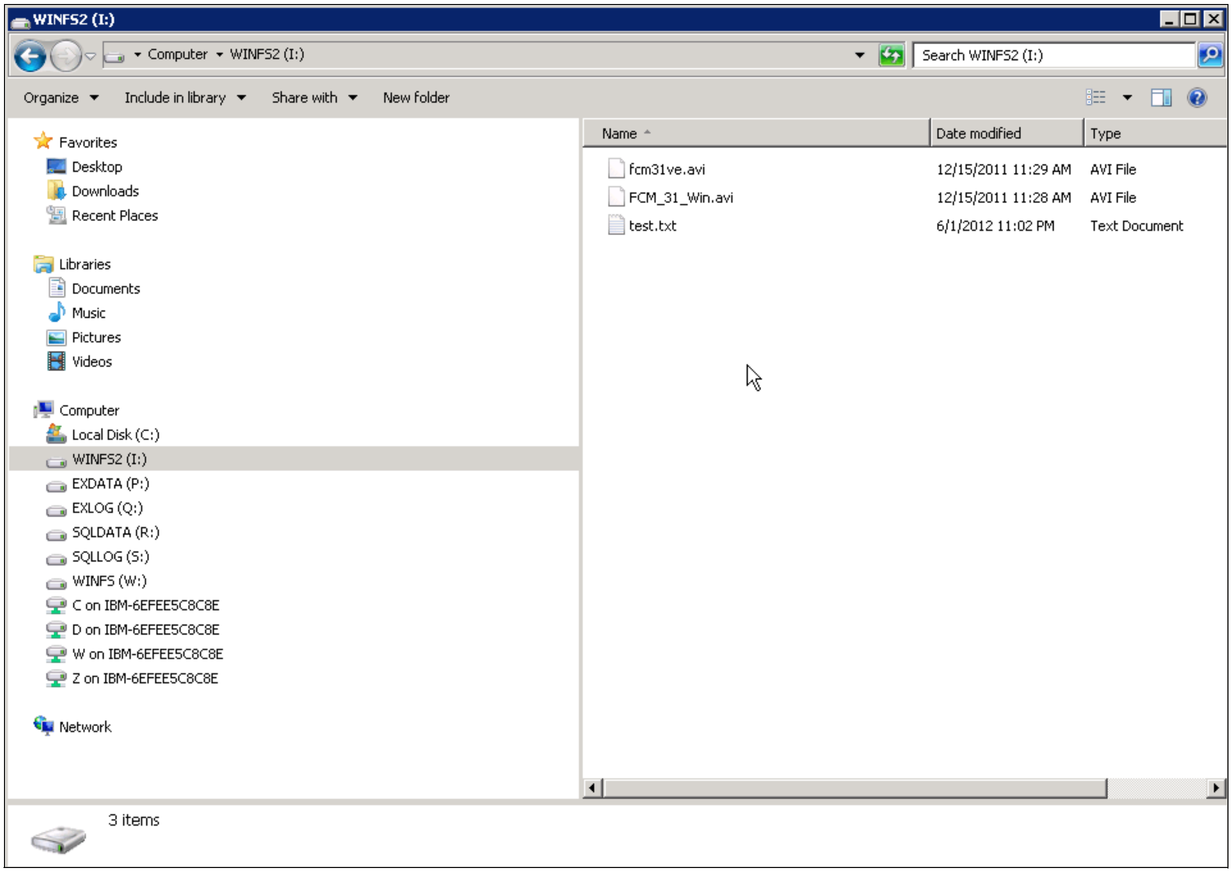


Figure 9-17 New disk drive mounted

Next, we create a new FlashCopy mapping at the SVC level as shown in Figure 9-18 on page 371.

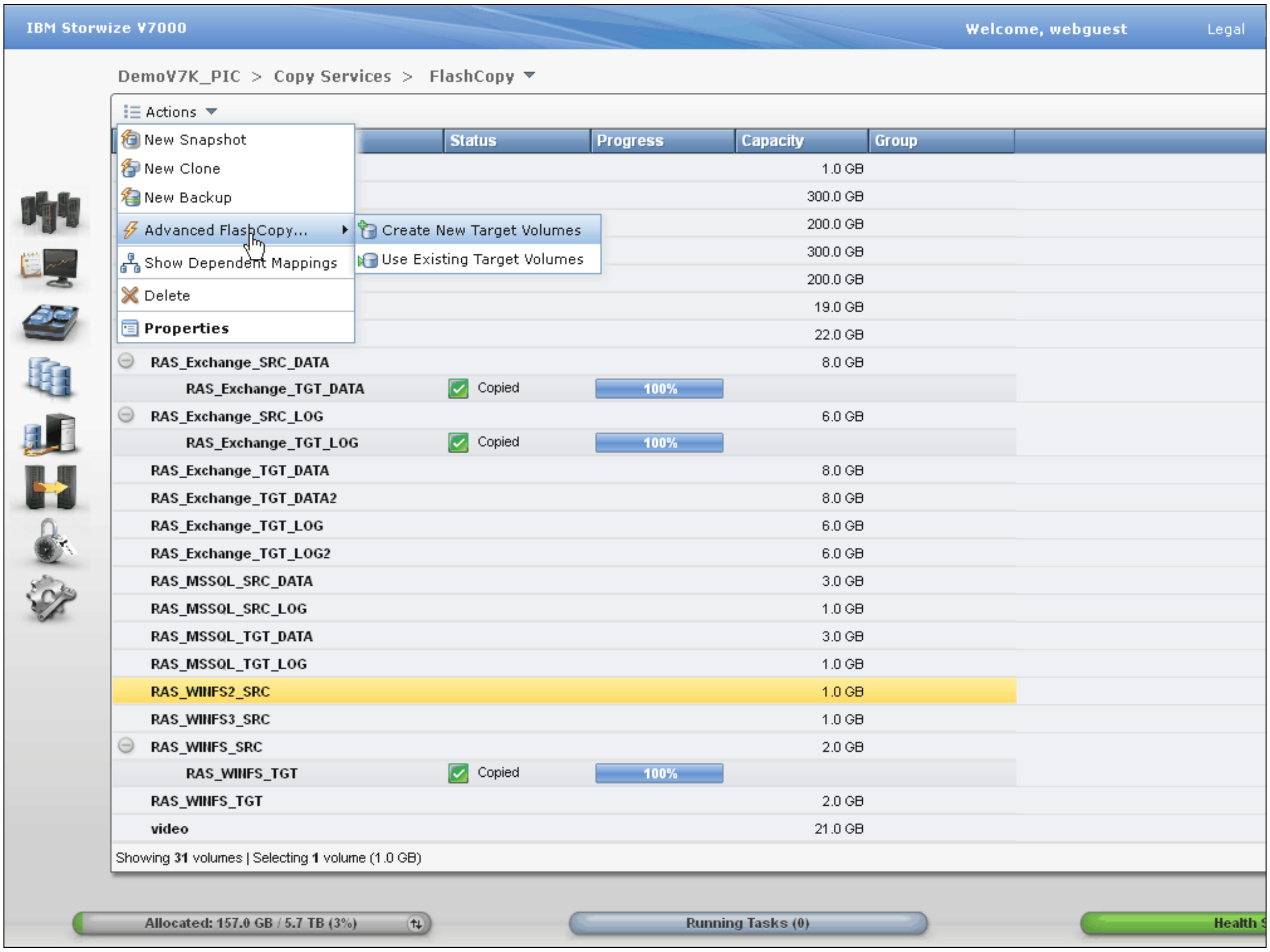


Figure 9-18 New FlashCopy mapping

The source volume RAS_WINFS2_SRC is already used for Disk 9 and mounted on I:. The target volume RAS_WINFS2_SRC_01 is a new volume with the same size and configuration. FlashCopy started and finished successfully as shown in Figure 9-19.

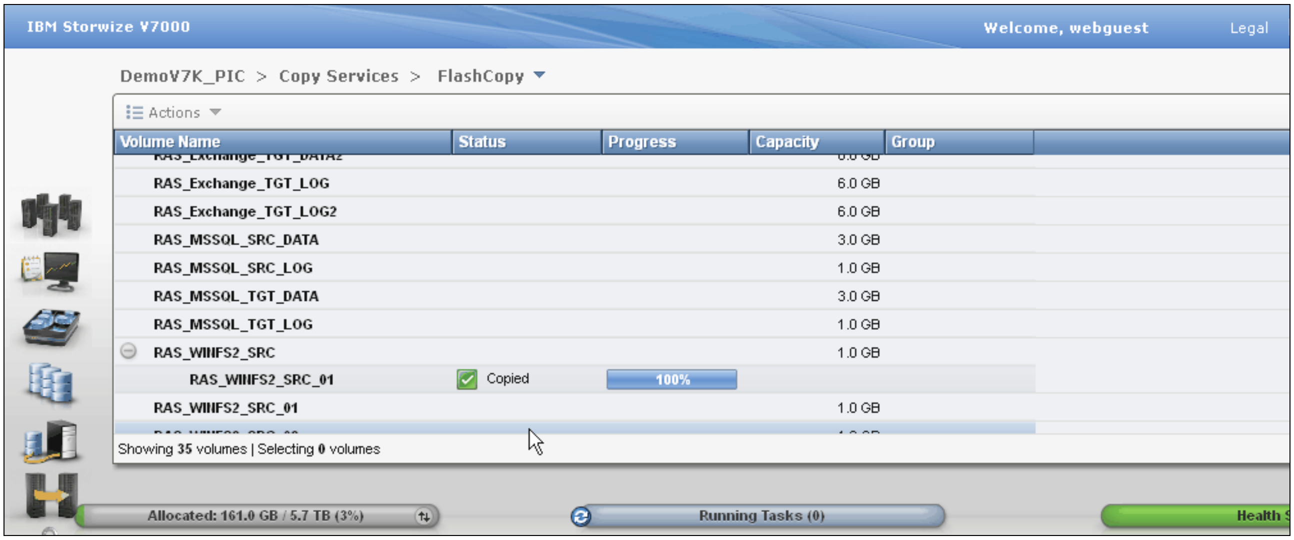


Figure 9-19 FlashCopy finished successfully

Mounting Basic Disk copy on the same host

We map the target volume to the server. Windows Server 2008 automatically discovers the new volume and it appears as Disk 11 Offline. We change it to `Online` and to Drive Letter `J:` as shown in Figure 9-20.

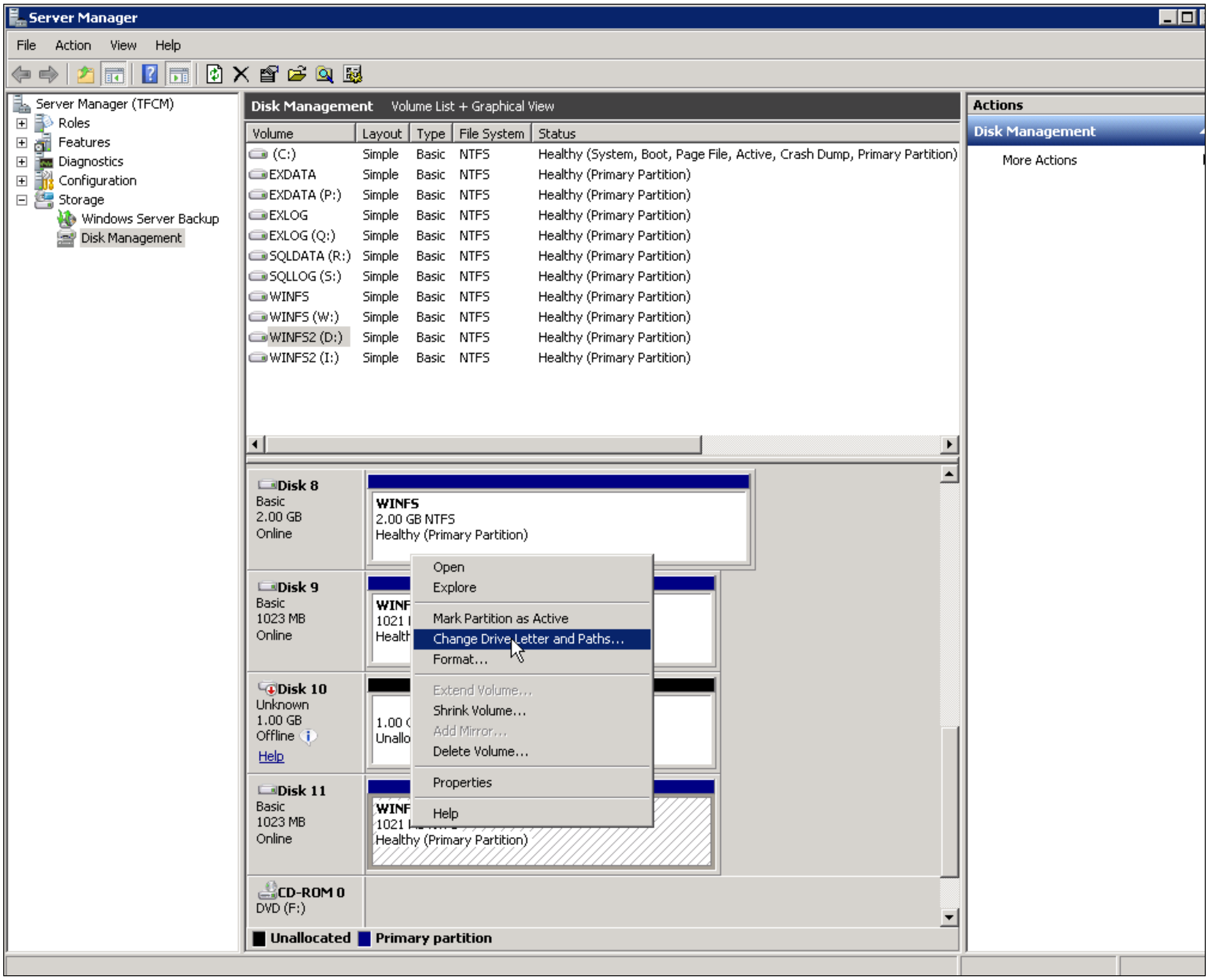


Figure 9-20 FlashCopy Target volume

And we get our Basic disk copy and files as shown in Figure 9-21 on page 374.

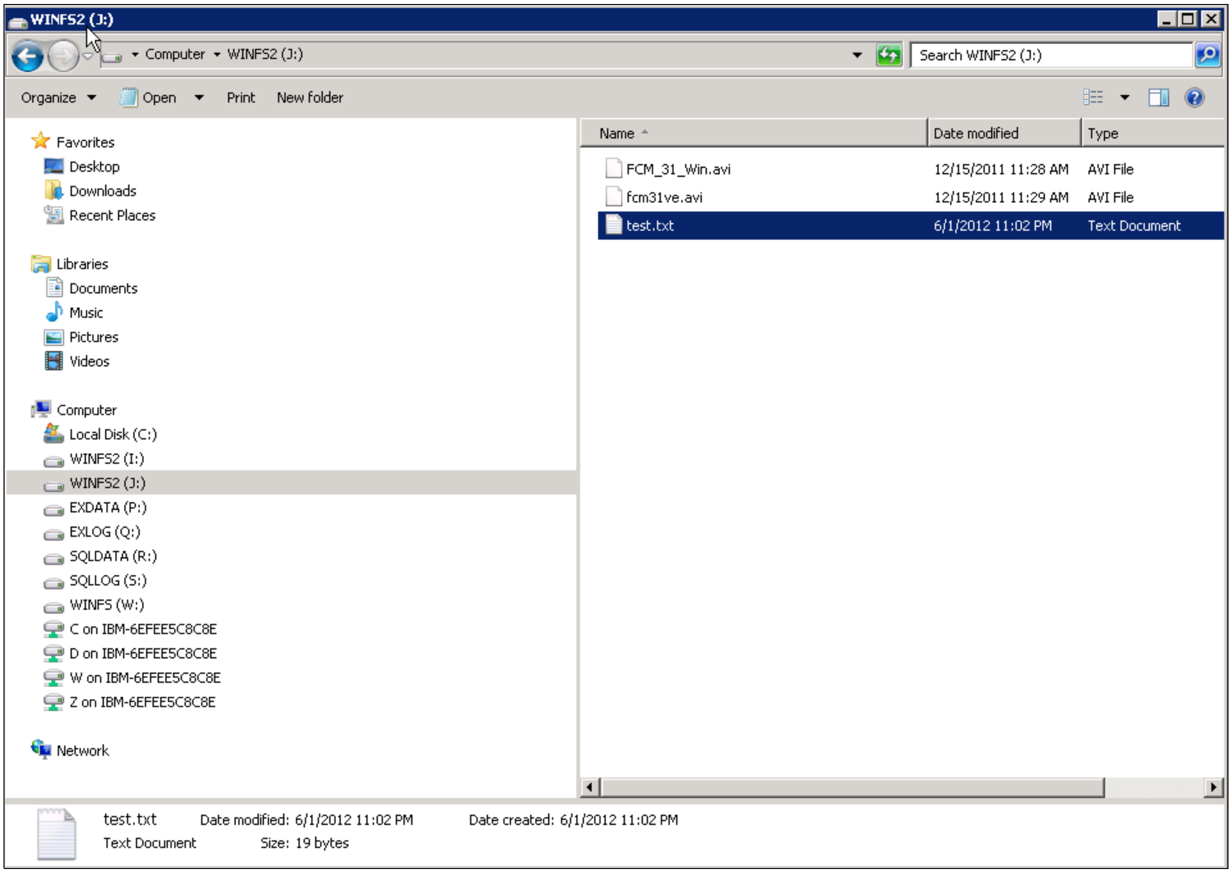


Figure 9-21 Basic disk copy and files

Basic Disk FlashCopy on the same host works in a straightforward manner, and it will also work in the same way if we mount it on another Windows Server 2008.

As you can see in Figure 9-22 on page 375, `Write-caching policy` is disabled in Windows Server 2008 and we have used Microsoft native driver with Windows Server 2008 and IBM SVC/Storwize V7000.

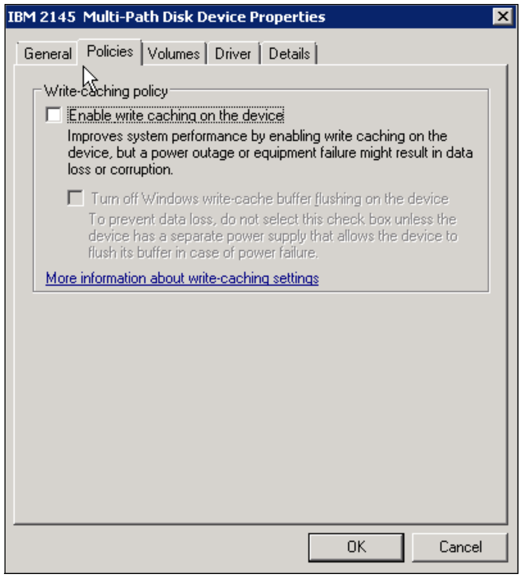


Figure 9-22 Write-caching policy is disabled

Creating Dynamic Disk copy

We repeat the same process, but the source volume is Dynamic Disk. We convert our Disk 9 to Dynamic Disk as shown in Figure 9-23 on page 376.

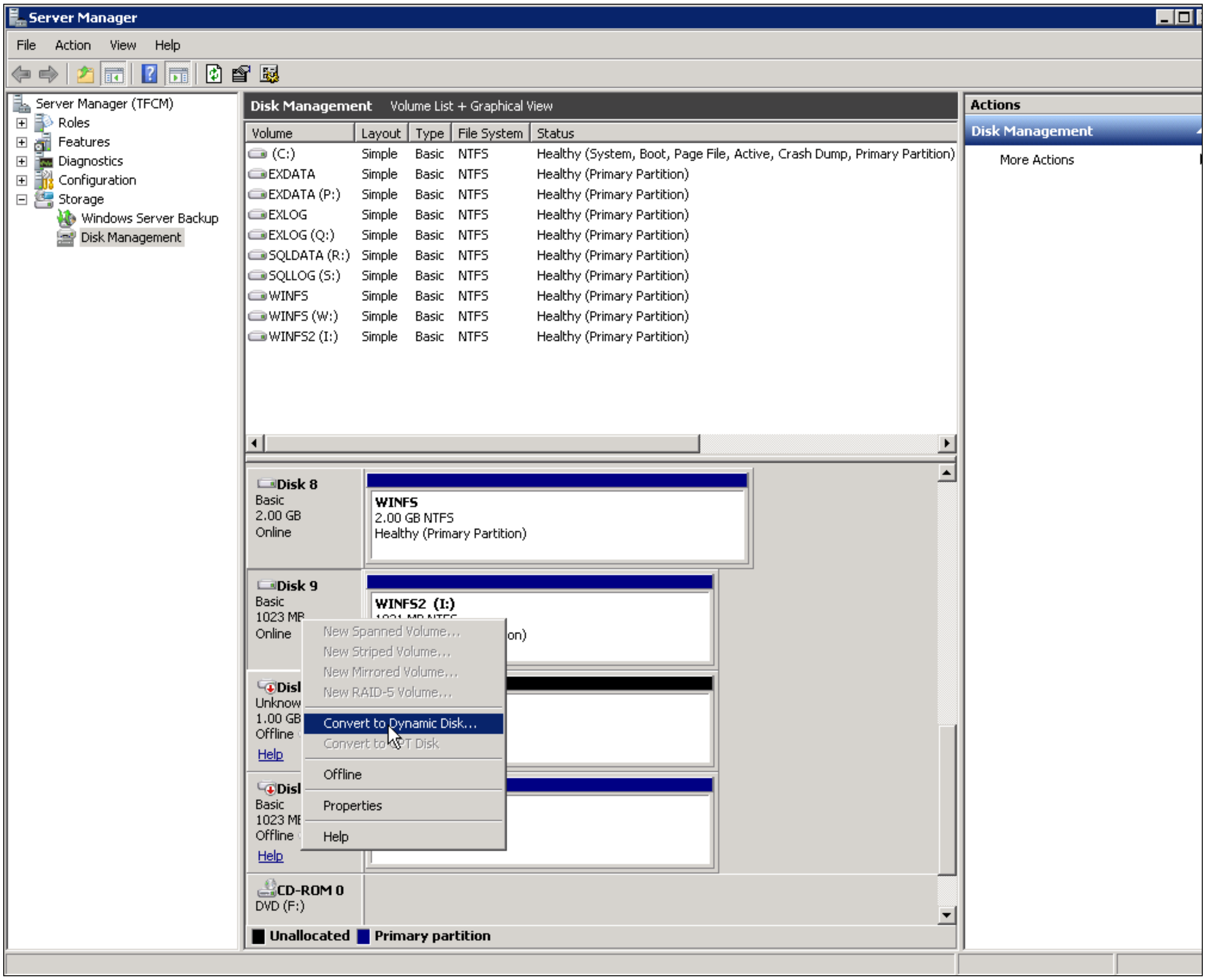


Figure 9-23 Convert to Dynamic Disk

We have mounted the I: volume with the same data as Dynamic Disk as shown in Figure 9-24 on page 377.

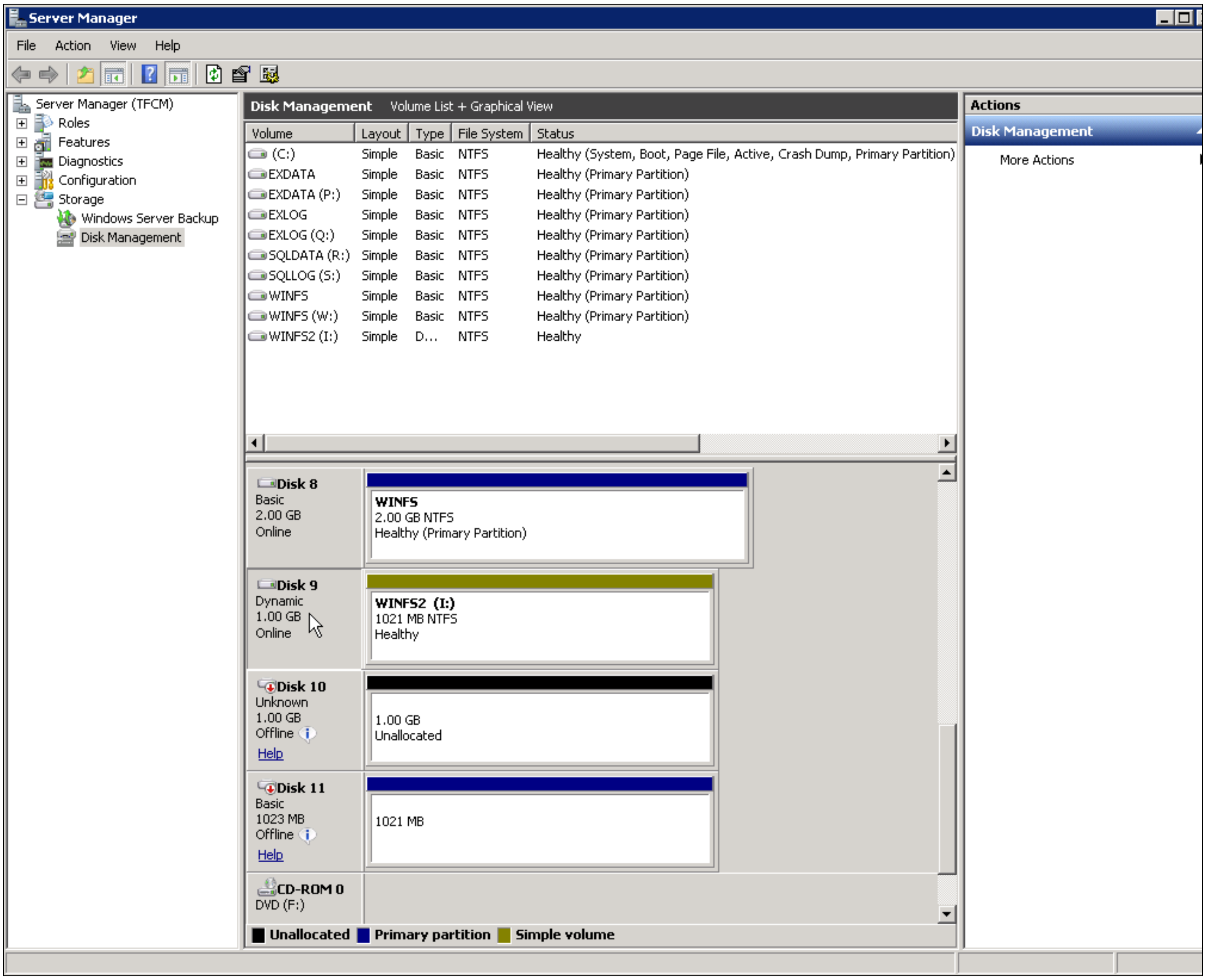


Figure 9-24 Dynamic Disk and data mounted

We create a new FlashCopy mapping and start it as shown in Figure 9-25 on page 378.

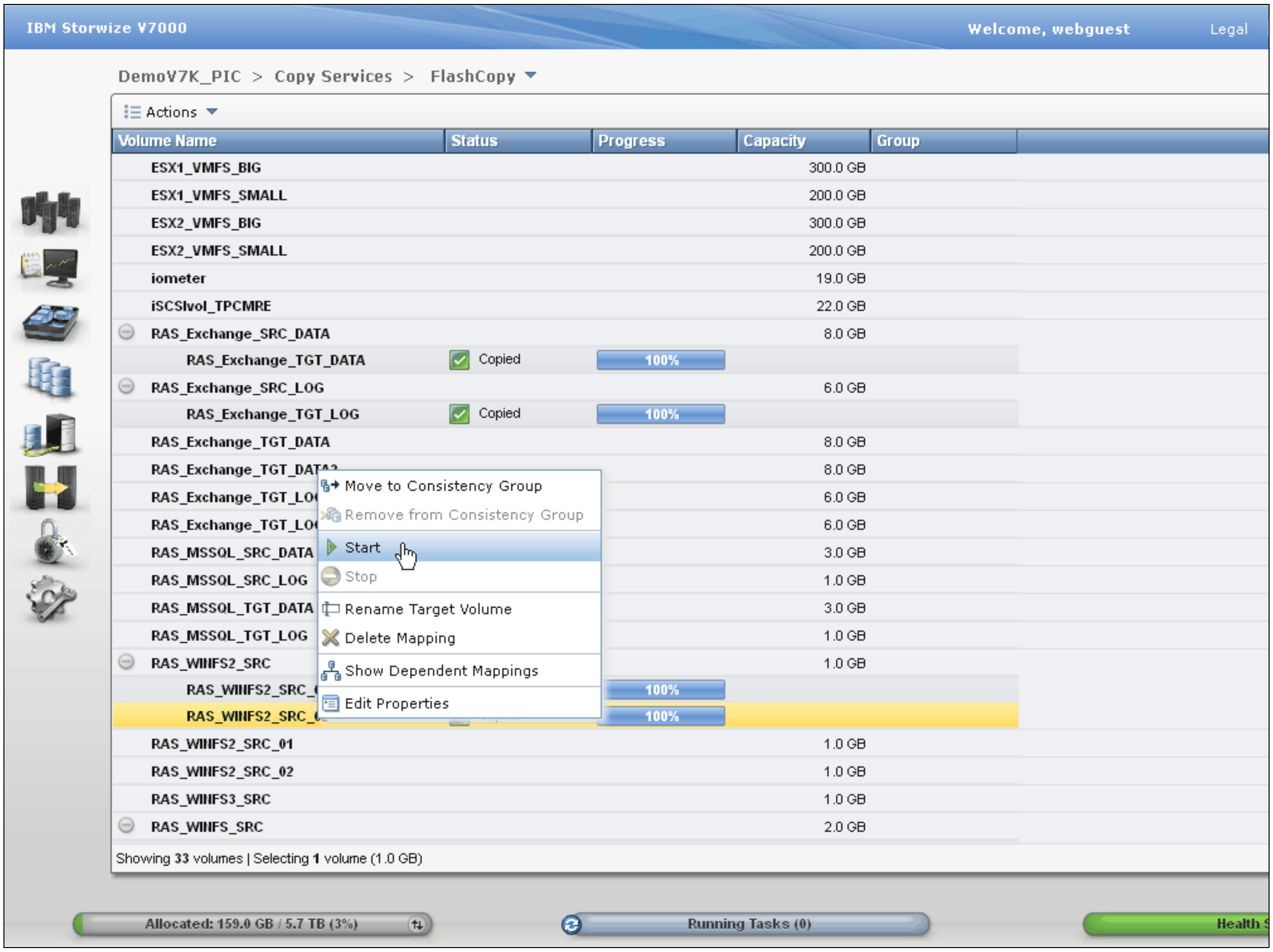


Figure 9-25 New FlashCopy

Mounting Dynamic Disk copy on the same host

FlashCopy Target volume is mapped to the host and Windows Server 2008 discovers it automatically as Disk 12. We attempt to change it to `Online` as shown in Figure 9-26 on page 379.

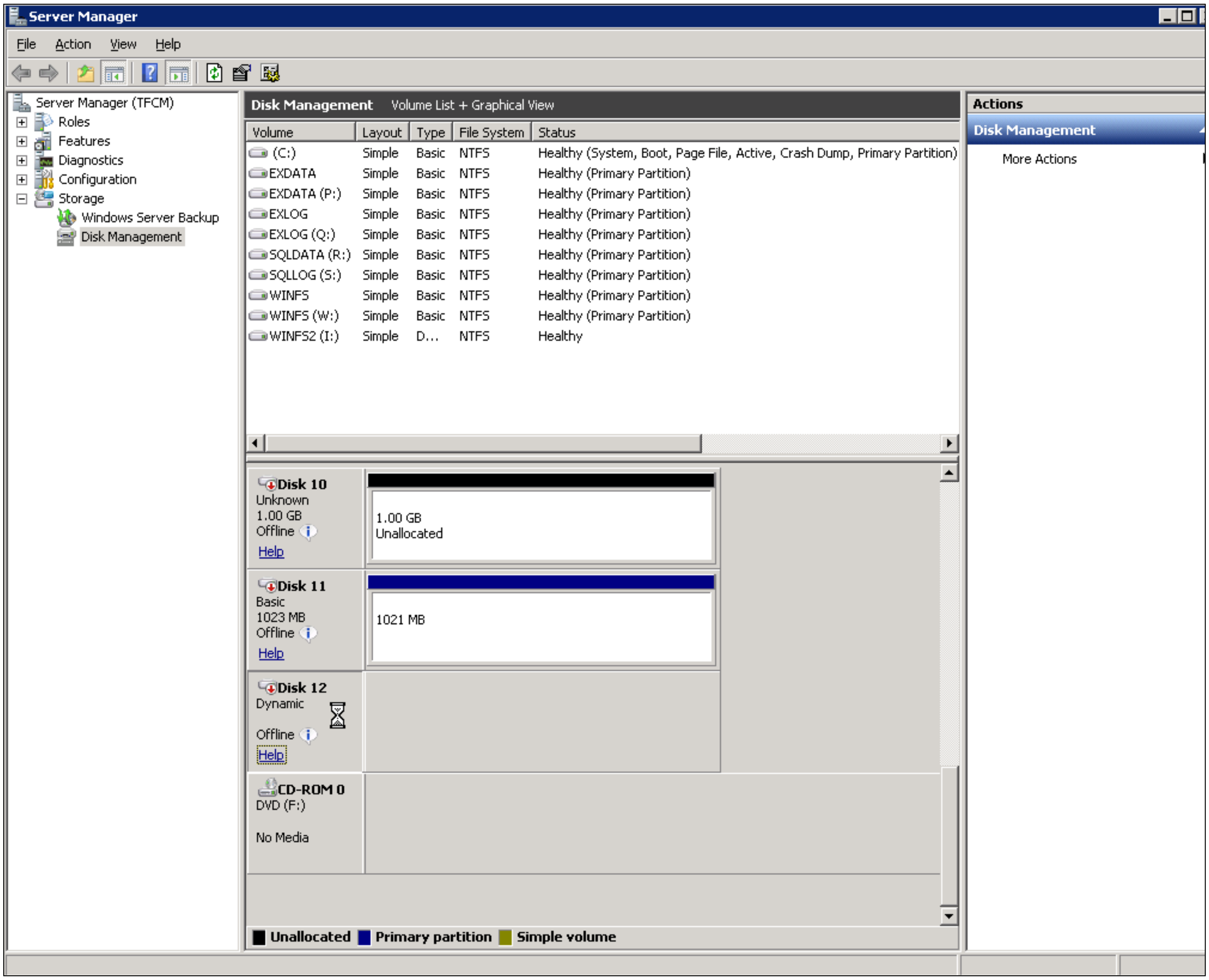


Figure 9-26 FlashCopy Target volume mapped

Instead, a status of `Invalid` is displayed, as shown in Figure 9-27 on page 380.

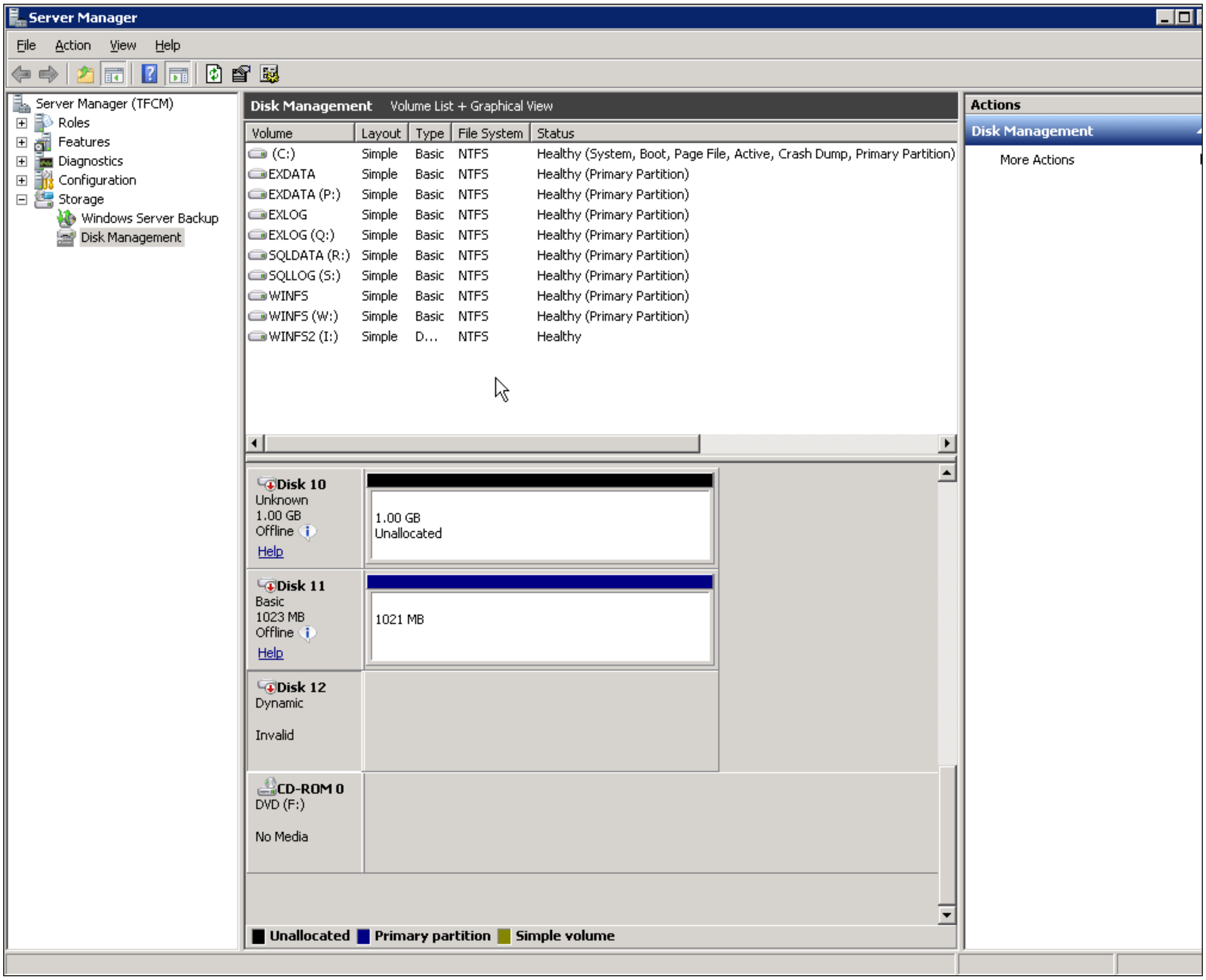


Figure 9-27 Dynamic Disks copy mounting is unsuccessful

Why did that happen? Because as previously mentioned, Dynamic Disks use a special part of the disk to maintain the LDM database. The LDM database contains disk management information like volume types, offsets, memberships, and drive letters of each volume. The LDM database replicates on all dynamic disks, so each dynamic disk knows about every other dynamic disk on this server. And when we try to use two dynamic disks with one and the same configuration it is unsuccessful. If we mount this Dynamic Disk target copy on another Windows Server 2008, it will work without any issues.

There are different options to change the LDM database in this case, but using IBM Tivoli Storage FlashCopy Manager for Windows is advisable. It is an advanced tool you can use to create and manage IBM SVC/Storwize V7000 Replication Family in the Microsoft Windows environment. See 11.2.1, "IBM Tivoli Storage FlashCopy Manager for Windows" on page 447, for more information about this topic.

9.5 Linux specifics

The steps needed to create and access copied volumes on Linux hosts are described here. SVC currently supports Red Hat Linux Advanced Server and SUSE Linux Enterprise Server, but the steps are distribution independent.

SVC supports the use of multiple multipathing solutions when accessed with a Linux server. The examples in the sections that follow focus on the use of SDD.

9.5.1 Creating useful copies

As with AIX and Windows, it is necessary to ensure that any information in the file system cache is written to the volumes prior to freezing the volumes. The simplest way to ensure that any information in the file system cache is written to the volumes prior to freezing the volumes is to unmount the file system, as shown in Example 9-1 on page 337.

The XFS file system includes support for freezing file systems in a similar manner to AIX. For more information about this topic, see XFS documentation.

Example 9-18 shows the initial configuration of the SVC cluster, presenting volumes to a Red Hat Linux server.

Example 9-18 Configuration of clusters prior to copying a volume with Metro Mirror

```
IBM_2145:ITSOCL2:admin>svcinfo lsvdisk -delim :
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
8:diomede-1ary:0:io_grp0:online:0:DS4500:36.0GB:striped:::::6005076801AD80E8E00000
0000000014:0

IBM_2145:ITSOCL2:admin>svcinfo lshost DIOMEDE
id 0
name DIOMEDE
port_count 2
type generic
mask 1111
iogrp_count 4
WWPN 210000E08B0548BC
node_logged_in_count 2
state active
WWPN 210000E08B0541BC
node_logged_in_count 2
state active


IBM_2145:ITSOCL3:admin>svcinfo lsvdisk -delim :
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
6:diomede-2dary:0:io_grp0:online:0:DS4500:36.0GB:striped:::::6005076801A100E908000
00000000008:0
```

Example 9-19 on page 382 shows the configuration of a Linux server prior to copying a volume with Metro Mirror.

Example 9-19 Linux host configuration prior to copying a volume with Metro Mirror

```
[root@diomede ~]# datapath query device

Total Devices : 1


DEV#:   0  DEVICE NAME: vpatha  TYPE: 2145     POLICY: Optimized Sequential
SERIAL: 6005076801ad80e8e000000000000014
============================================================================
Path#      Adapter/Hard Disk       State     Mode          Select     Errors
    0      Host0Channel0/sda        OPEN   NORMAL               1          0
    1      Host0Channel0/sdb        OPEN   NORMAL           41815          0
    2      Host1Channel0/sdc        OPEN   NORMAL               0          0
    3      Host1Channel0/sdd        OPEN   NORMAL           41679          0


[root@diomede ~]# pvdisplay
  --- Physical volume ---
  PV Name               /dev/hda2
  VG Name               VolGroup00
  PV Size               74.41 GB / not usable 0
  Allocatable           yes
  PE Size (KByte)       32768
  Total PE              2381
  Free PE               3
  Allocated PE          2378
  PV UUID               Xq0UQs-94ID-tCq7-4jK2-KRt7-4iYF-lCf9wf

  --- Physical volume ---
  PV Name               /dev/vpatha
  VG Name               itso
  PV Size               36.00 GB / not usable 0
  Allocatable           yes
  PE Size (KByte)       4096
  Total PE              9215
  Free PE               4607
  Allocated PE          4608
  PV UUID               dAZQYq-VsI1-n2xp-vp0c-hfSE-glCx-MjnPV3


[root@diomede ~]# vgdisplay itso
  --- Volume group ---
  VG Name               itso
  System ID
  Format                lvm2
  Metadata Areas        1
  Metadata Sequence No  2
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                1
  Open LV               1
  Max PV                0
  Cur PV                1
  Act PV                1
```

```
  VG Size               36.00 GB
  PE Size               4.00 MB
  Total PE              9215
  Alloc PE / Size       4608 / 18.00 GB
  Free  PE / Size       4607 / 18.00 GB
  VG UUID               OPsMcJ-nDr8-b3ds-YyCM-Mlde-OegZ-LZFddT


[root@diomede /]# df -T
File System                      Type  1K-blocks Used     Available Use% Mounted
on
/dev/mapper/VolGroup00-LogVol00 ext3  74699952  11571300 59334120  17% /
/dev/hda1                       ext3  101086    22817    73050     24% /boot
none                            tmpfs 1033144   0        1033144   0%  /dev/shm
/dev/mapper/itso-mmPrimary      ext3  18578172  77888    17556568  1%  /mmSource


[root@diomede /]# mount
/dev/mapper/VolGroup00-LogVol00 on / type ext3 (rw)
none on /proc type proc (rw)
none on /sys type sysfs (rw)
none on /dev/pts type devpts (rw,gid=5,mode=620)
usbfs on /proc/bus/usb type usbfs (rw)
/dev/hda1 on /boot type ext3 (rw)
none on /dev/shm type tmpfs (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
/dev/mapper/itso-mmPrimary on /mmSource type ext3 (rw)
```

The VG that is copied is `itso`. This VG contains a single vpath, `vpatha`.

There are no user commands for freezing this `ext3` file system, so the only way to freeze the volume cleanly is to unmount the file system and deactivate the LV. However, with appropriate planning and scripting, the disruption can be minimized. Follow these steps:

1. Create the Metro Mirror Consistency Group.

   ```
   IBM_2145:ITSOCL2:admin>svctask mkrcconsistgrp -name diomedeMM -cluster ITSOCL3
   RC Consistency Group, id [253], successfully created
   ```

2. Create the Metro Mirror relationship connecting the two volumes and add it to the Consistency Group.

   ```
   IBM_2145:ITSOCL2:admin>svctask mkrcrelationship -master diomede-1ary -aux
   diomede-2dary -cluster ITSOCL3 -name diomedeMM1 -consistgrp diomedeMM
   RC Relationship, id [8], successfully created
   ```

3. Start the Consistency Group.

   ```
   IBM_2145:ITSOCL2:admin>svctask startrcconsistgrp diomedeMM
   ```

4. Monitor the Consistency Group and wait until it attains consistency and synchronization.

   ```
   IBM_2145:ITSOCL2:admin>svcinfo lsrcconsistgrp diomedeMM
   id 253
   name diomedeMM
   master_cluster_id 000002006B603A38
   master_cluster_name ITSOCL2
   aux_cluster_id 0000020068403A42
   aux_cluster_name ITSOCL3
   ```

```
primary master
state consistent_synchronized
relationship_count 1
freeze_time
status online
sync
copy_type metro
RC_rel_id 8
RC_rel_name diomedeMM1
```

5. Unmount the file system to flush all the data to the primary volume.

```
[root@diomede ~]# umount /mmSource
```

6. Stop the Metro Mirror Consistency Group. This is the freeze point:

```
IBM_2145:ITSOCL3:admin>svctask stoprcconsistgrp -access diomedeMM

IBM_2145:ITSOCL3:admin>svcinfo lsrcconsistgrp diomedeMM
id 253
name diomedeMM
master_cluster_id 000002006B603A38
master_cluster_name ITSOCL2
aux_cluster_id 0000020068403A42
aux_cluster_name ITSOCL3
primary
state idling
relationship_count 1
freeze_time
status
sync in_sync
copy_type metro
RC_rel_id 6
RC_rel_name diomedeMM1
```

7. Remount the file system:

```
[root@diomede ~]# mount /dev/itso/mmPrimary /mmSource
```

At this point, the Metro Mirror relationship has been stopped and `diomede-2ary` is an exact copy of `diomede-1ary`.

9.5.2 Accessing useful copies

As with AIX, if the source or primary volume is defined to the AIX logical volume manager (LVM), all of its data structures and identifiers are copied to the target or secondary volume as well. This information includes the Universal Unique Identifiers (UUIDs) that are used to identify PVs, VGs, and LVs.

It is currently not possible to activate a volume group with a physical volume (vpath) that contain a VG UUID and a PV UUID that are already used in a volume group existing on the same server.

9.5.3 Accessing target or secondary volumes from the same Linux host

It is possible to access target or secondary volumes from the same Linux host; however, accessing target or secondary volumes from the same Linux host requires manually editing the LVM data stored on the disk to assign a different UUID. Methods to assign a different

UUID vary by Linux distribution and version, so precise commands and scripts are beyond the intended scope of this book.

Attention: This method of accessing disks is at your own risk and is not supported by IBM.

9.5.4 Accessing target or secondary volumes from a different Linux host

This section describes how to recover a volume that was *not* frozen cleanly, which can happen when a Metro Mirror relationship stops while a file system is in use. Follow these steps:

1. Map the target volume to the new server:

```
IBM_2145:ITSOCL3:admin>svctask mkvdiskhostmap -host PALAU diomede-2dary
Virtual Disk to Host map, id [0], successfully created
```

2. Discover the logical unit (LU) on the Linux server. QLogic provides a tool to perform this step. You can obtain more information at:

   http://filedownloads.qlogic.com/files/ms/69917/readme_LUN_disc_Linux.html

3. Run **cfgvpath** to create the relevant vpath.

```
[root@palau ~]# cfgvpath
c---------  1 root root 254, 0 Nov 14 22:32 /dev/IBMsdd
Added vpatha 252 0       path sda 8 0....
Added vpatha 252 0       path sdb 8 16....
Added vpatha 252 0       path sdc 8 32....
Added vpatha 252 0       path sdd 8 48....
Writing out new configuration to file /etc/vpath.conf
Waiting for hotplug/udev system to configure all vpath device nodes...
```

4. Run **pvscan** to pick up the copied PV.

```
[root@palau ~]# pvscan
  PV /dev/hda2     VG VolGroup00   lvm2 [74.41 GB / 96.00 MB free]
  PV /dev/vpatha   VG itso         lvm2 [36.00 GB / 18.00 GB free]
  Total: 2 [110.40 GB] / in use: 2 [110.40 GB] / in no VG: 0 [0   ]
```

5. Run **vscan** to make sure that you have your VG.

```
[root@palau ~]# vgscan
  Reading all physical volumes.  This may take a while...
  Found volume group "VolGroup00" using metadata type lvm2
  Found volume group "itso" using metadata type lvm2
```

6. Run **lvscan** to make sure that you have your expected LVs.

```
[root@palau ~]# lvscan
  ACTIVE            '/dev/VolGroup00/LogVol00' [72.38 GB] inherit
  ACTIVE            '/dev/VolGroup00/LogVol01' [1.94 GB] inherit
  inactive          '/dev/itso/mmPrimary' [18.00 GB] inherit
```

7. You must activate your newly discovery volume group.

```
[root@palau ~]# vgchange -ay itso
  1 logical volume(s) in volume group "itso" now active
```

8. Run **fsck** on the LV and then mount it.

9.6 Other operating systems

The principles outlined in the previous sections are equally relevant to other operating systems. Be aware of the following considerations, however, when determining operating procedures:

- What metadata is written to the disk (and thus copied by copy services)?
- How does the operating system handle duplicated metadata?
- How does the operating system handle “foreign” metadata, that is, disks that were originally mapped to a different server?
- How does the operating system handle volumes that were not frozen cleanly?

10

Automating copy services processes

As a general rule, system designs require a human operator's involvement at a certain stage. For example, human beings need to decide when to stop and start FlashCopy mappings or Remote Copy relationships, and when to fail over or fail back Remote Copy relationships.

The object of *automation* is to remove the human factor from performing systems operations. For your copy services solution, there are various approaches that you can use to automate processes, including the following methods:

- SAN Volume Controller (SVC) command-line scripting
- Server-side scripting
- SVC Common Information Model (CIM) Object Manager (CIMOM) programming

These methods are discussed here along with logging, which is an important part of automation. The following topics are covered:

- Running commands on the cluster
- Creating connections
- SVC command-line scripting
- SVC CIMOM programming
- Logging
- Auditing

10.1 Running commands on the cluster

To automate copy services processes, you need to connect to the cluster. In normal operations, you connect to the cluster by using the GUI or command line. The GUI is not an appropriate interface for automating processes, so that alternative is not discussed here. All automation techniques are achieved through the SVC command line or the Common Information Model Object Manager (CIMOM), which currently acts as a proxy to the command line.

This section uses the term *user agent*. The user agent can be the CIMOM, which connects to the cluster by using Secure Shell (SSH). Or the user agent can be a user connecting directly with an SSH client, either in an interactive mode or by means of a script.

Running commands to the cluster follows these steps:

1. Connection
2. Authentication
3. Submission
4. Authorization
5. Execution

The following section discusses these steps in more detail.

10.1.1 Connection

Commands are submitted to the cluster during a connection session to the cluster. User agents make connections through the SSH protocol. SVC has a number of security features that affect how often you can attempt connections. These security features are in place to prevent attacks (malicious or accidental) that can bring down an SVC node. These features might initially seem restrictive, but they are relatively simple to work around to maintain a valid connection.

Any automation system must ensure that it behaves responsibly and does not attempt to breach the connection rules. At a minimum, an automation system must ensure that it can gracefully handle rejected connection attempts.

Figure 10-1 shows how SVC connection restrictions work.

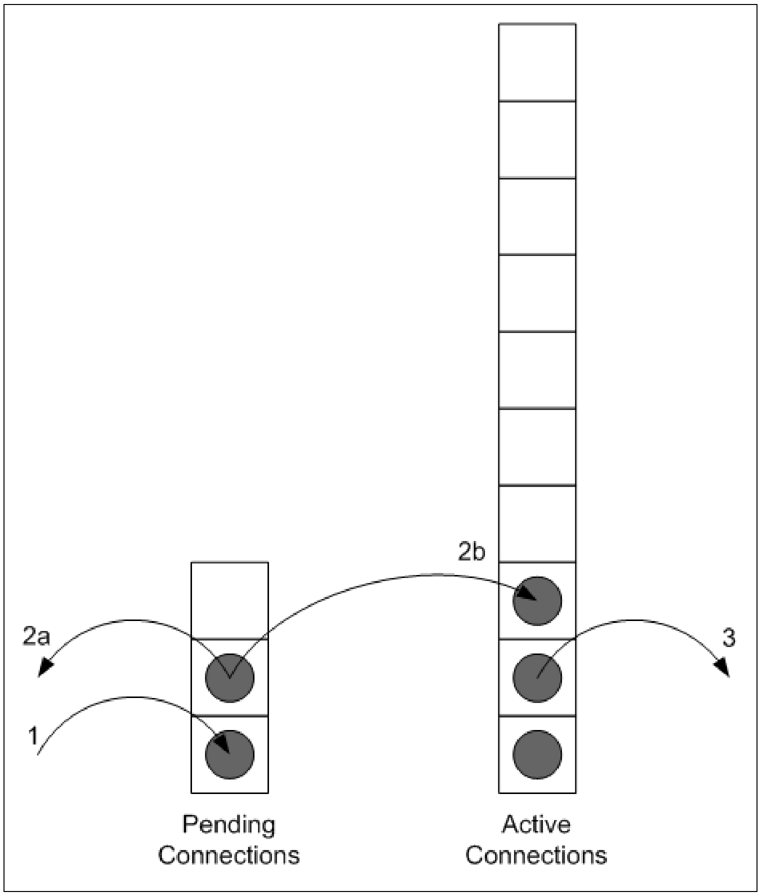


Figure 10-1 SVC SSH restrictions

In Figure 10-1, two “queues” are in action: Pending Connections and Active Connections. The connection process follows this sequence:

1. A connection request comes into the SVC. If the Pending Connections queue has a free position, the request is added to it. Otherwise, there are two possibilities. That is, prior to Version 4.2.1.0, the connection request times out. And as of Version 4.2.1.0, the connection is explicitly rejected.
2. Pending Connections are handled in one of two ways:
   a. If any of the following conditions are true, the connection request is rejected:
      - No key is provided, or the provided key is incorrect.
      - The provided user name is not *admin* or *service*.
      - The Active Connections queue is full.

      In this case, a warning is returned to the SSH client as shown in Example 10-1 on page 390.
   b. If none of the conditions listed in the previous step are true, the connection request is accepted and moved from the Pending Connections queue to the Active Connections queue.

3. Active Connections end after any of the following events:
   - The user logs off manually.
   - The SVC SSH daemon recognizes that the connection has grown idle.
   - Network connectivity fails.
   - The configuration node fails over.

     In this case, both queues are cleared as the SHH daemon stops and restarts on a different node.

Example 10-1 SVC command-line warning about too many logins

```
login as: admin
Authenticating with public key "rsa-key-20080728"
Too many logins for 'admin'.
Last login: Tue Aug 12 21:25:42 2008 from 9.164.158.255
```

Only 10 concurrent active SSH sessions are allowed. If this limit is reached, an entry is generated on the error log as shown in Figure 10-2.

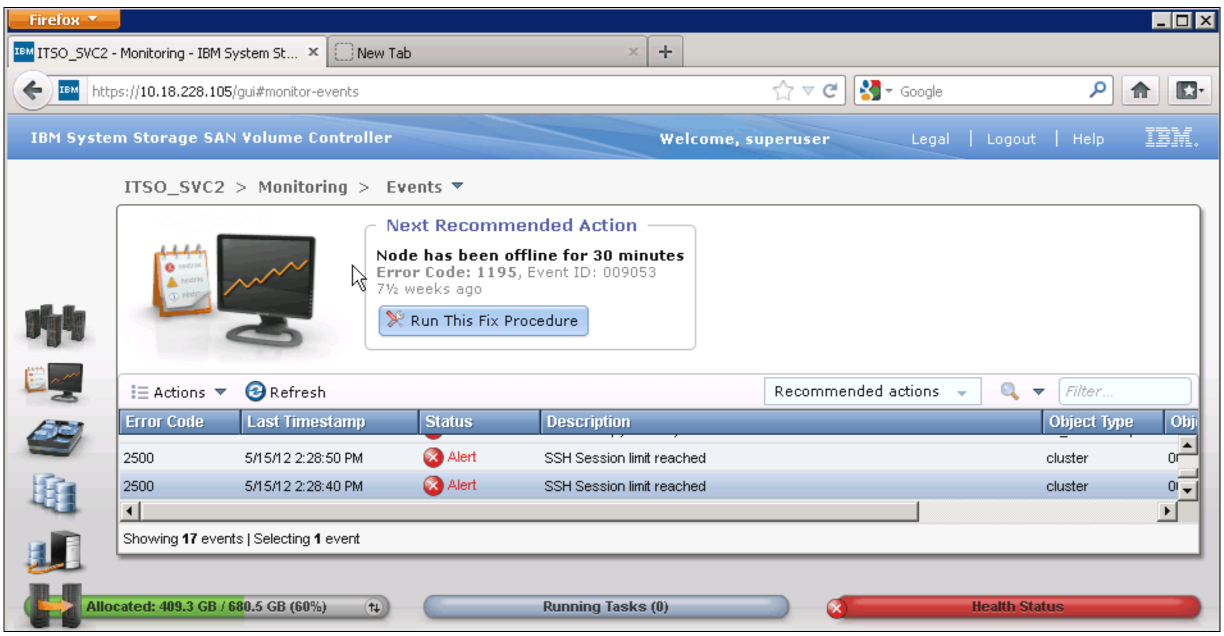


Figure 10-2 Error 2500 - SSH Session limit reached

To view the details, right-click the event and choose **Properties**. Event details are displayed as shown in Figure 10-3 on page 391.

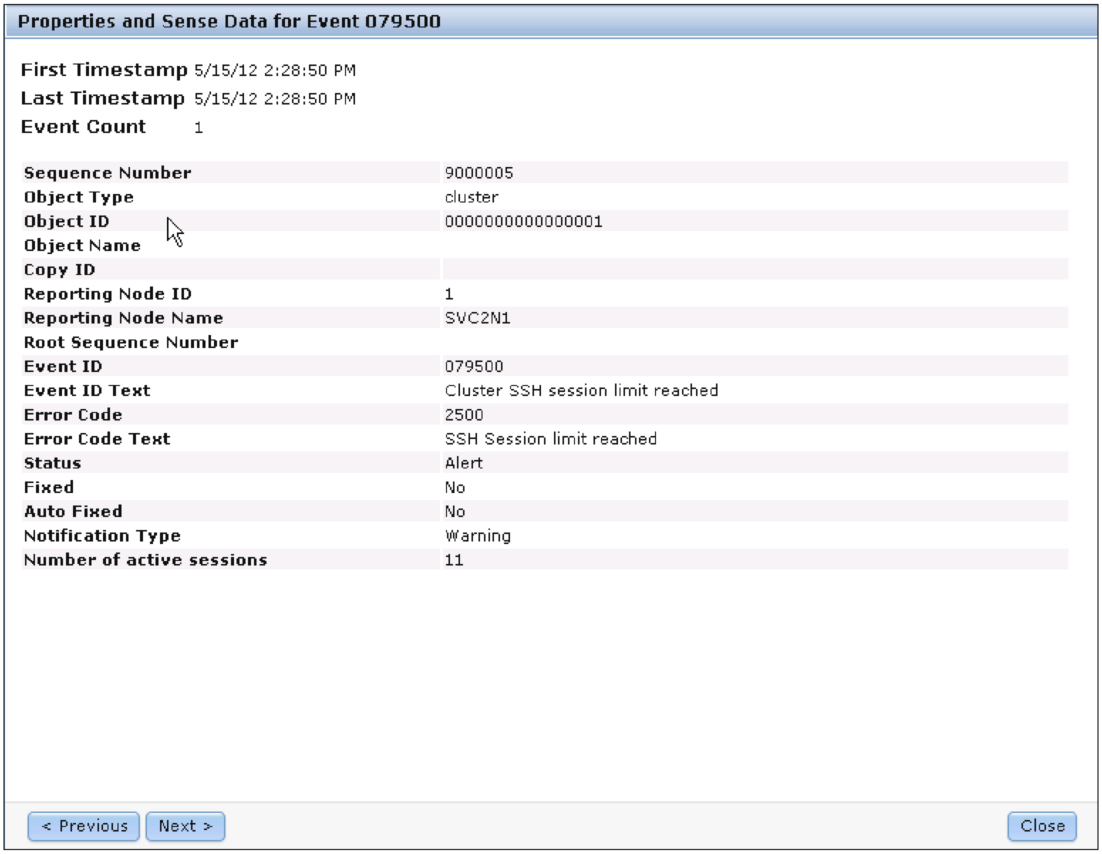


Figure 10-3 Event details

To fix this error, right-click the event and choose the **Run Fix Procedure**; a list with active SSH sessions is displayed.

At bottom of the list is a **Close all SSH sessions** option that you can use if you do not have a more suitable way to close the active sessions. Figure 10-4 on page 392 displays an example of 10 active sessions.

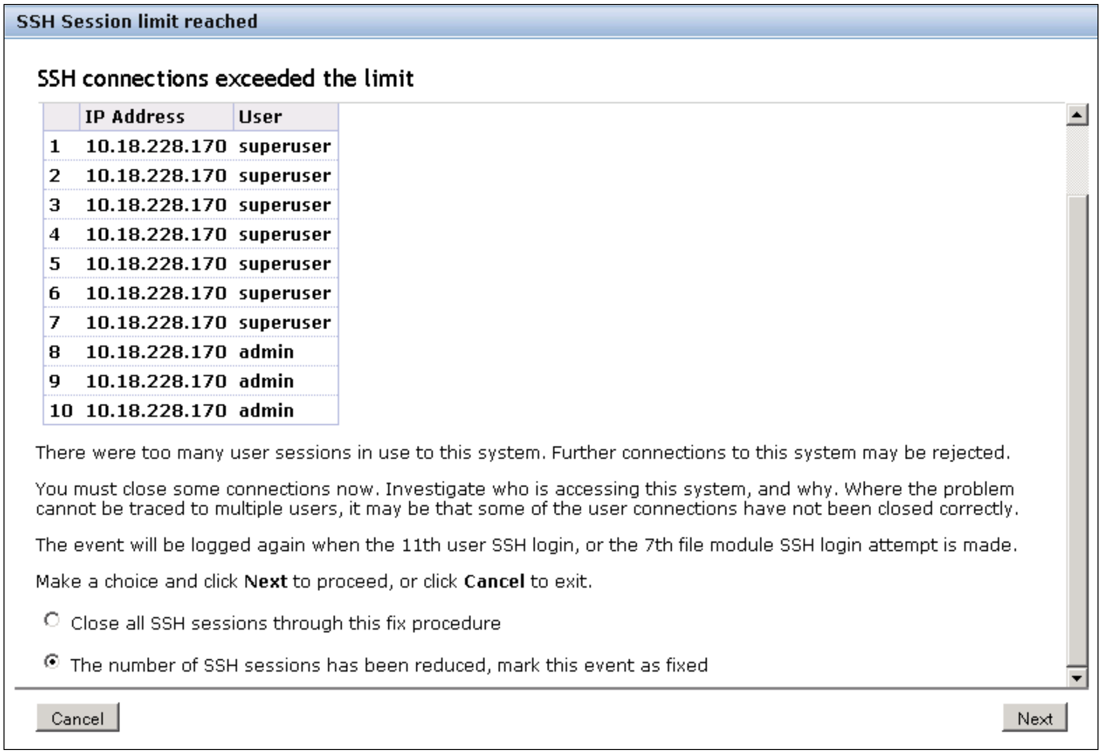


Figure 10-4 List of active SSH sessions

Clicking **Close all SSH sessions** closes the listed sessions, and the error is fixed. If you close the active sessions manually on the host side without choosing to close all of the sessions through the **Run Maintenance Procedures**, you must select **The number of SSH sessions has been reduced**.

10.1.2 Authentication

Starting with Version 6.3 you can login with simply a user name and password. You no longer need a Secure Shell (SSH) key.

There are two types of users who can access the system: local users and remote users. These types are based on how the users are authenticated to the system.

Local users must provide a password, an SSH key, or both. Local users are authenticated through the authentication methods that are located in the SAN Volume Controller system. If the local user needs access to the management GUI, a password is needed for the user. If the user requires access to the command-line interface (CLI) through SSH, either a password or a valid SSH key file is necessary. Local users must be part of a user group that is defined on the system. User groups define roles that authorize the users within that group to a specific set of operations on the system.

Remote users are authenticated on a remote service with either Tivoli Integrated Portal or Lightweight Directory Access Protocol (LDAPv3) support such as IBM Tivoli Storage Productivity Center or IBM Tivoli Directory Server. A remote user does not need local authentication methods. With Tivoli Integrated Portal, both a password and SSH key are required to use the command-line interface. With LDAP, having a password and SSH key is not necessary, although SSH keys optionally can be configured. Remote users who need to access the system when the remote service is down also need to configure local credentials. Remote users have their groups defined by the remote authentication service.

To manage users and user groups on the system using the management GUI, select **User Management** → **Users**. To configure remote authentication with Tivoli Integrated Portal or Lightweight Directory Access Protocol, select **Settings** → **Directory Services**.

Refer to 10.7, “Auditing” on page 424 for a discussion on the auditing of commands on the SVC cluster.

10.1.3 Submission

When connected to a cluster, the user agent can start submitting commands. First, the syntax is checked. If the syntax checking fails, an appropriate error message is returned. Any automation implementation must ensure that all submitted commands have the correct syntax. If they do not, they must be designed to handle syntax errors. It is much easier to design a solution that does not generate invalid syntax than it is to design a solution to handle all potential syntax errors.

10.1.4 Authorization

Next, commands with valid syntax are checked to determine whether the user agent has the authority to submit the command. A role is associated with the key that was used to authenticate the connection. SVC checks the submitted command against the authorization role. If the user agent is not authorized to execute this command, the following error is returned: `CMMVC6253E` The task has failed because the user's role is not authorized to submit the command. If the user agent is authorized, the command is sent for execution.

For an in-depth discussion on authorization and roles refer to the following IBM Red books:

- *Implementing the IBM System Storage SAN Volume Controller V6.3*, SG24-7933
- *Implementing the IBM Storwize V7000 V6.3*, SG24-7938

10.1.5 Execution

When a command is executed, one of the following actions occurs:

- The command fails, and an error message is written to STDERR.
- The command succeeds, and a warning is written to STDERR.
- The command succeeds, a warning is written to STDERR, and information is sent to STDOUT.
- The command succeeds, and information is written to STDOUT.
- The command succeeds, and nothing is written to STDOUT.

Data that is written to STDOUT and STDERR by the cluster is written to STDOUT and STDERR by your SSH client. However, you must verify that the data was written to STDOUT and STDERR by your SSH client yourself.

10.2 Creating connections

Connecting to the cluster is the first step in running commands. Any automation solution requires a connection component. This component must be as robust as possible, because it forms the foundation of your solution.

There are two forms of connection solutions:

- Transient

  One command is submitted per connection, and the connection is closed after the command is completed.

- Persistent

  The connection is made and stays open. Multiple commands are submitted through this single connection, including interactive sessions and the CIMOM.

10.2.1 Transient connections

Transient connections are simple to create. The most common SSH clients enable the user to submit a single command as part of the user's invocation. Example 10-2 shows a user submitting a single command as part of the user's invocation using **ssh** on an AIX server.

Example 10-2 Transient connection to SVC cluster from AIX server

```
root@Kanaga:/>ssh -i privateKey -l admin itsosvccl2 svcinfo lscluster -delim :
id:name:location:partnership:bandwidth:cluster_IP_address:cluster_service_IP_addre
ss:cluster_IP_address_6:cluster_service_IP_address_6:id_alias
0000020066403A44:itsosvccl2:local:::9.43.86.119:9.43.86.120:::0000020066403A44
0000020063A03A38:itsosvccl3:remote:fully_configured:50:9.43.86.131:9.43.86.132:::0
000020063A03A38
root@Kanaga:/>
```

Example 10-3 shows a user submitting a single command as part of the user's invocation using **plink** on a Windows server.

Example 10-3 Transient connection to SVC cluster from Windows server

```
C:\Program Files\PuTTY>plink -i "c:\Documents and Settings\Administrator\ITSO\priv
ate.ppk" -l admin itsosvccl2 svcinfo lscluster -delim :
id:name:location:partnership:bandwidth:cluster_IP_address:cluster_service_IP_addre
ss:cluster_IP_address_6:cluster_service_IP_address_6:id_alias
0000020066403A44:itsosvccl2:local:::9.43.86.119:9.43.86.120:::0000020066403A44
0000020063A03A38:itsosvccl3:remote:fully_configured:50:9.43.86.131:9.43.86.132:::0
000020063A03A38

C:\Program Files\PuTTY>
```

These transient connections go through all five stages of running a command and return to the command line. You can redirect the two output streams (STDOUT and STDERR) using the operating system's standard redirection operators to capture the responses.

These lengthy invocations can be shorted in client-specific ways. User configuration files can be used with the AIX SSH client. The configuration file in Example 10-4 on page 394 enables you to create a transient connection.

Example 10-4 Sample SSH configuration file saved as "sampleCfg"

```
Host cl2
HostName itsosvccl2
IdentityFile /privateKey
User admin
```

```
Host someOtherCluster
IdentityFile /someOtherKey
User admin
```

The transient connection is shown in Example 10-5.

Example 10-5 Transient connection to SVC cluster using ssh and configuration file

```
root@Kanaga:/>ssh -F sampleCfg cl2 svcinfo lscluster -delim :
id:name:location:partnership:bandwidth:cluster_IP_address:cluster_service_IP_addre
ss:cluster_IP_address_6:cluster_service_IP_address_6:id_alias
0000020066403A44:itsosvccl2:local:::9.43.86.119:9.43.86.120:::0000020066403A44
0000020063A03A38:itsosvccl3:remote:fully_configured:50:9.43.86.131:9.43.86.132:::0
000020063A03A38
root@Kanaga:/>
```

Shortening the plink invocation requires the creation of a PuTTY session. First, open the PuTTY application and enter `admin@<`*`Host Name`*` or `*`cluster IP address`*`>` in the Host Name (or IP address) field as shown in Figure 10-5.

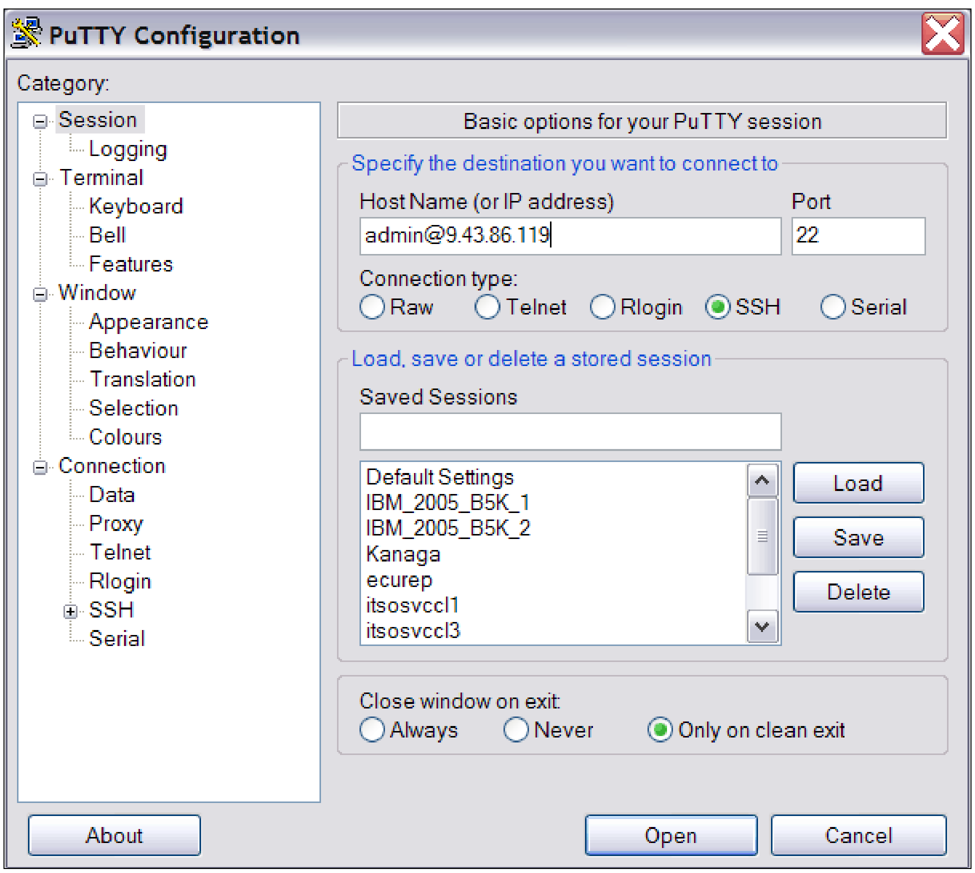


Figure 10-5 Add user name and IP address to a PuTTY session

As shown in Figure 10-6, the next step is to configure the private key for this session. Select **Connection** → **SSH** → **Auth**.

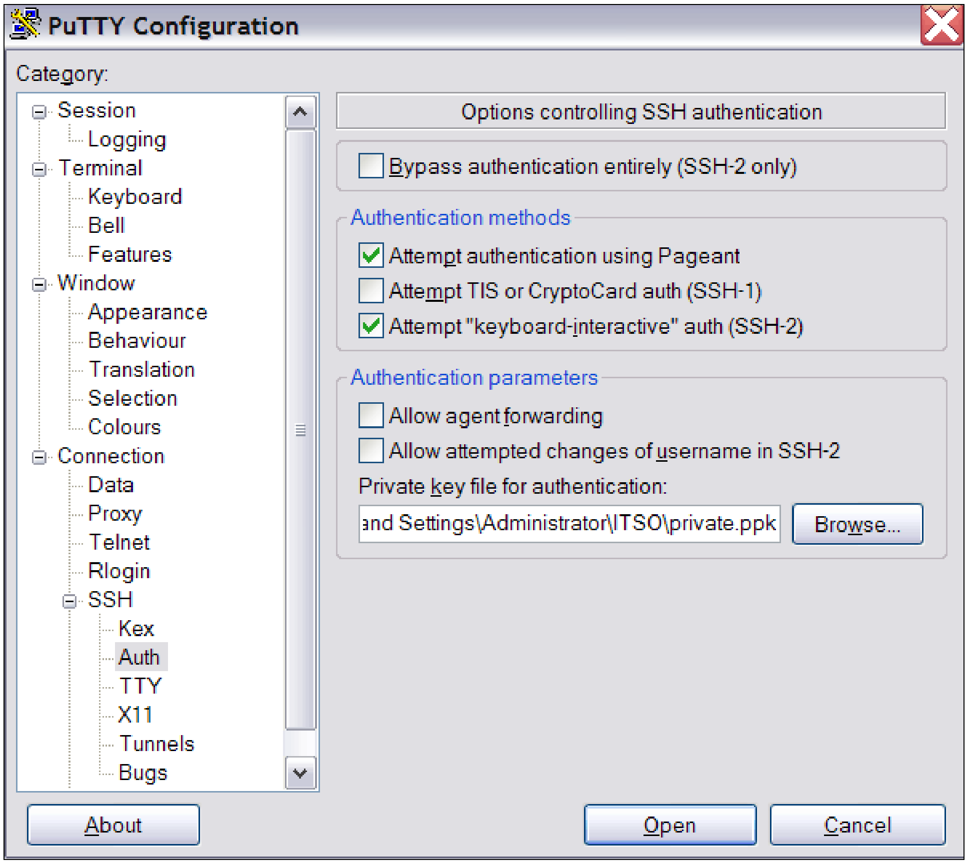


Figure 10-6 Set private key for a PuTTY SSH session

Click **Browse** and select the appropriate private key by locating it in your file system. The final step is to save the session for future use. Use a descriptive session name as shown in Figure 10-7 on page 397.

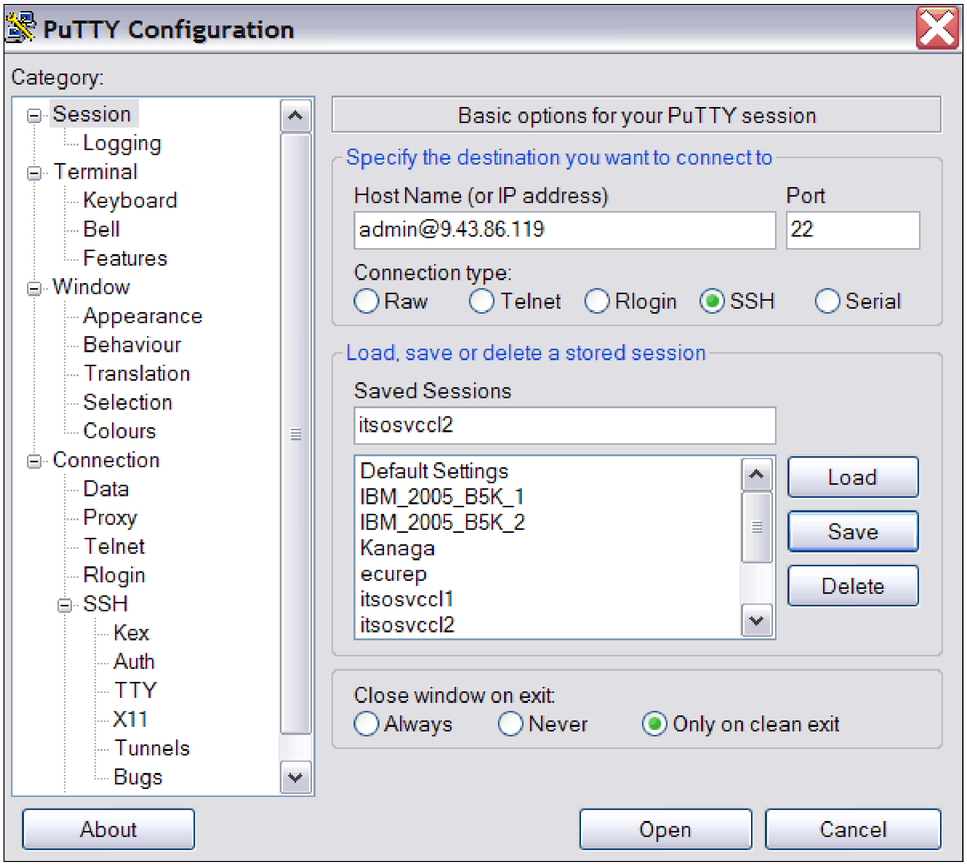


Figure 10-7 Save PuTTY session for use with plink

After a session has been saved, you can use it to make transient connections from the command line (Example 10-6).

Example 10-6 Transient connection to an SVC cluster using plink with a PuTTY session

```
C:\Program Files\PuTTY>plink -load itsosvccl2 svcinfo lsnode
id              name              UPS_serial_number     WWNN             status
       IO_group_id     IO_group_name      config_node       UPS_unique_id    hardware

1               node1             100068A008            5005076801001D22 online
       0               io_grp0            yes               2040000188440008 8F4

2               node2             100068A006            5005076801001D21 online
       0               io_grp0            no                2040000188440006 8F4


C:\Program Files\PuTTY>
```

Example 10-7 shows a transient connection from a UNIX or Linux host to perform an SVC configuration backup.

Example 10-7 Transient connection to SVC cluster using ssh on UNIX or Linux host

```
[root@Demo backup]# ssh superuser@10.18.228.105 svcconfig clear -all
..
CMMVC6155I SVCCONFIG processing completed successfully
[root@Demo backup]#

[root@Demo backup]# ssh superuser@10.18.228.105 svcconfig backup
................................................................................
CMMVC6155I SVCCONFIG processing completed successfully
[root@Demo backup]#

[root@Demo backup]# scp superuser@10.18.228.105:/tmp/svc.config.backup.* .
svc.config.backup.log                                                     100%
24KB  23.8KB/s   00:00
svc.config.backup.sh                                                      100%
11KB  11.1KB/s   00:00
svc.config.backup.xml                                                     100%
269KB 268.8KB/s   00:00
[root@Demo backup]#
```

There are times when system administrators want to obtain details for the volumes mapped to their servers; new volumes have been created, or you need a full list with volumes mapped to this host. Currently there is no tool or command to give you this information from IBM SVC/Storwize V7000 systems. However, you can use the following example to quickly obtain information about a specific host and volume from any host with SSH (a similar method applies to hosts with PuTTy and plink).

First, create a new user with the Monitor role. Go to **Access** → **Users** → **Monitor** and click **New User** as shown in Figure 10-8.

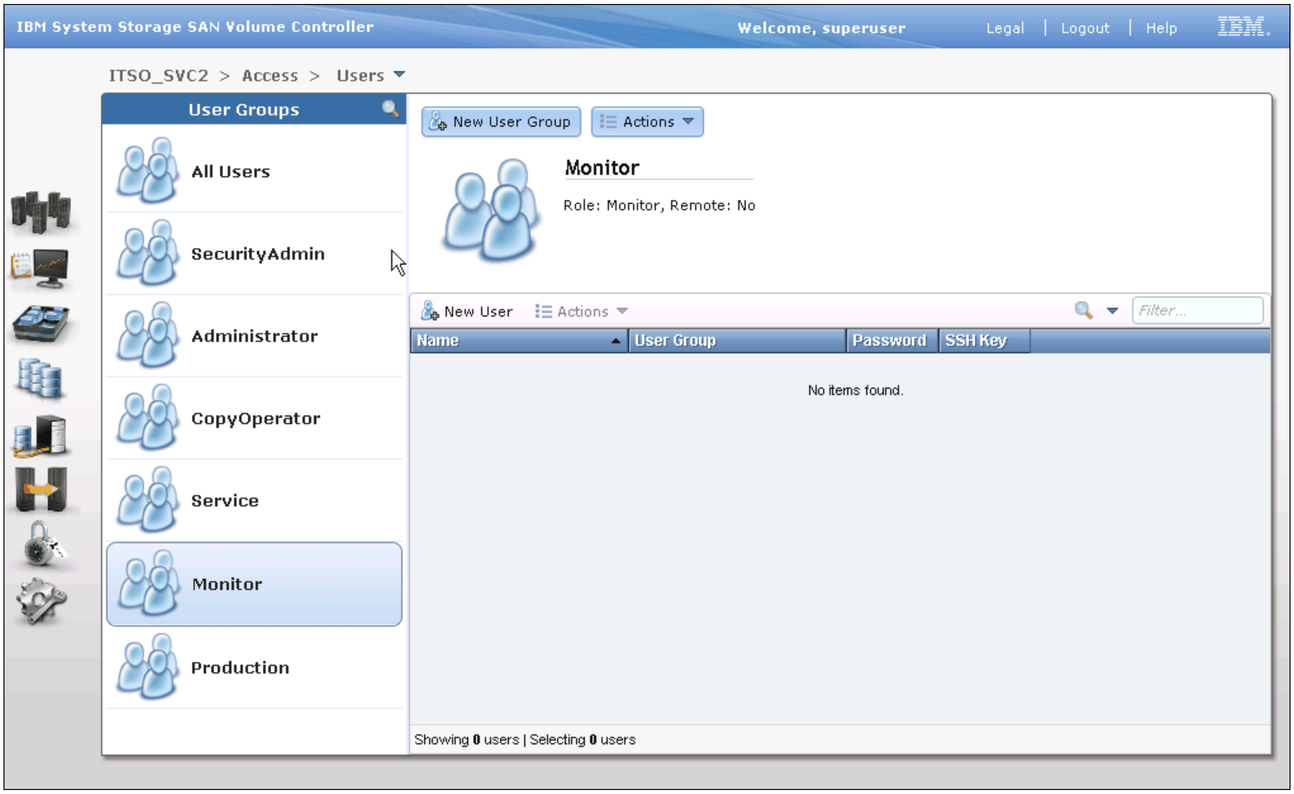


Figure 10-8 Monitor role

Fill in the details and click **Create** (Figure 10-9 on page 399).

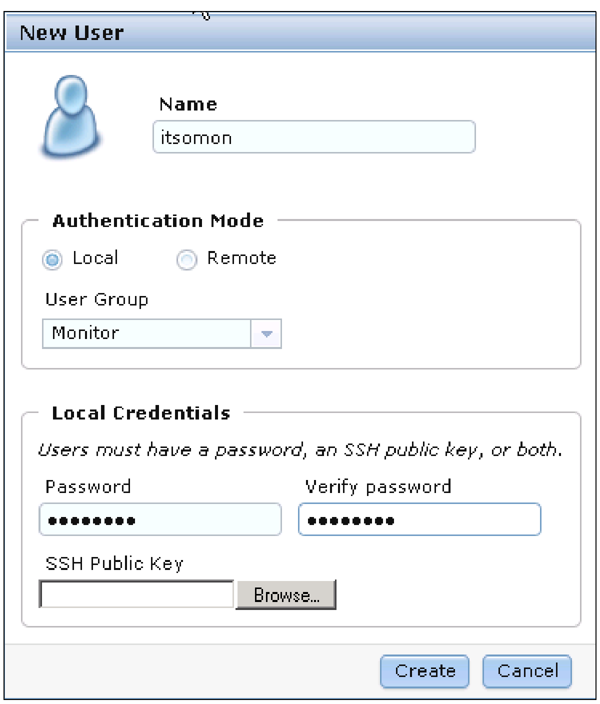


Figure 10-9 User details

You can see the command that is actually running in the background, as shown in Figure 10-10. This is a useful feature in the SVC and Storwize V7000 Graphical User Interface (GUI) because every time you perform changes to the storage, you can see the CLI command that has been created. This is helpful if you need to automate any process or run it a number of times.

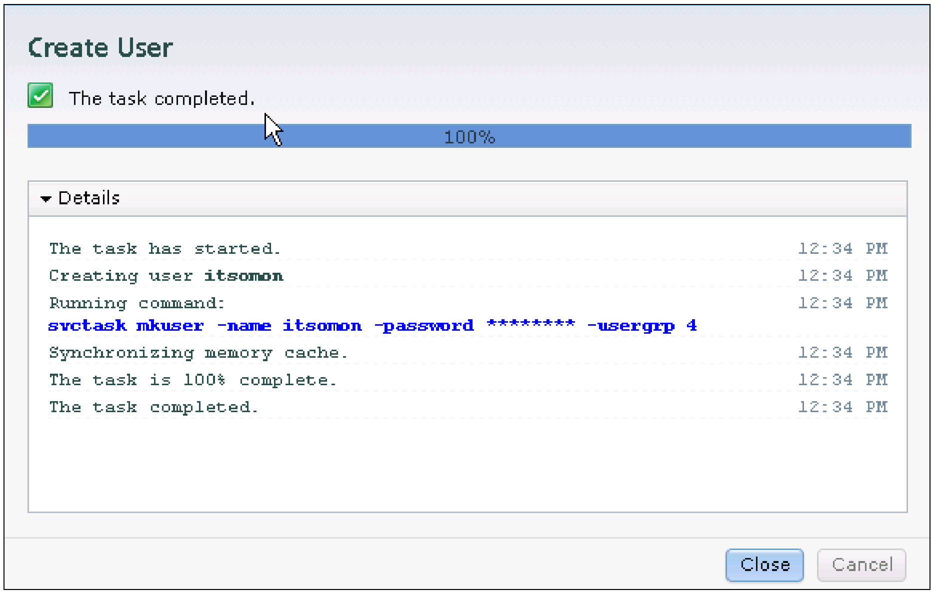


Figure 10-10 Running command in background

You now have a new user, as shown in Figure 10-11 on page 400.

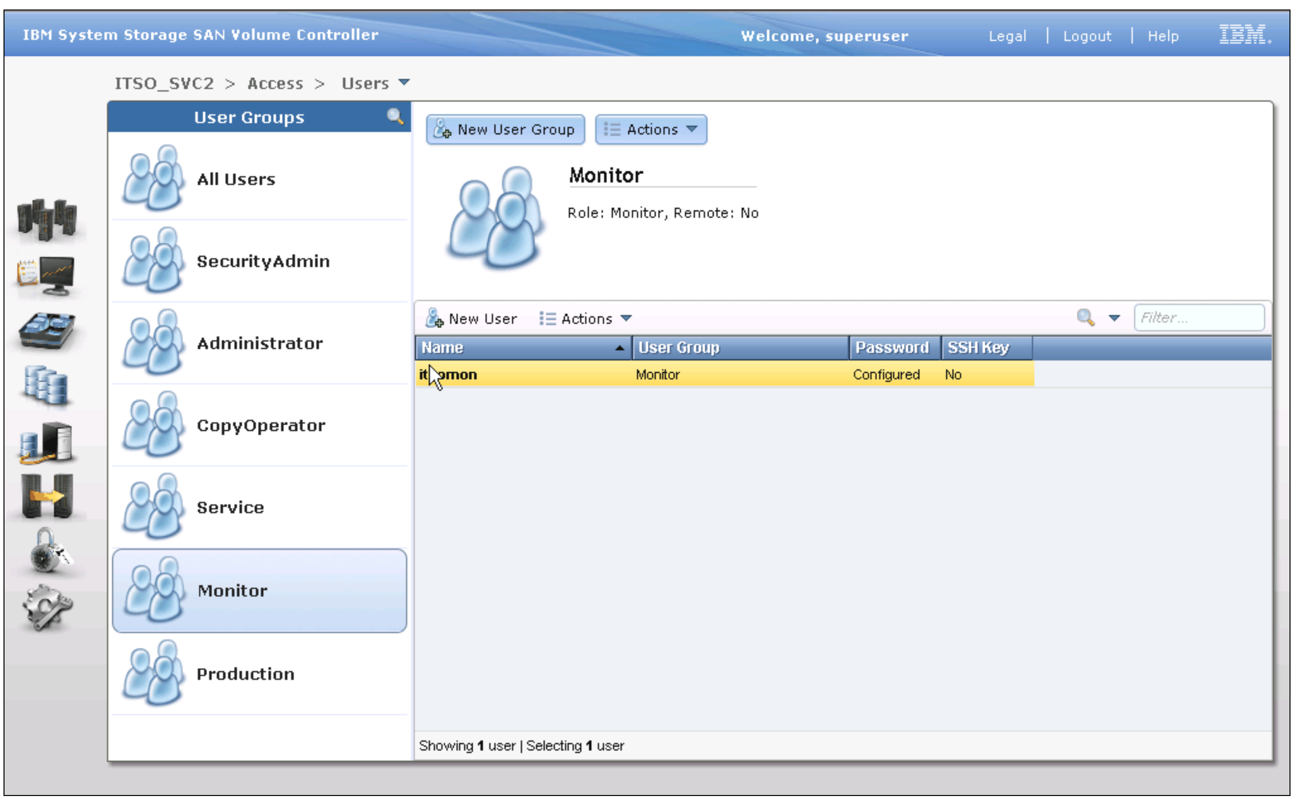


Figure 10-11 New user with the role of Monitor

The Monitor role provides access to list commands only. This user cannot perform actions that change the state of the system. So you now have a user that can be utilized to obtain storage information only, which is ideal for monitoring purposes.

From the server side, any system administrator with this user access can run several list commands and obtain a list with all hosts, volumes, and mappings on a particular SVC cluster, filters for a particular host, and obtain the status and capacity for particular volumes. Example 10-8 is an example using the **lshostvdiskmap** command.

Example 10-8 List commands and filters

```
[root@Demo ~]# ssh itsomon@10.18.228.105 lshostvdiskmap
id name              SCSI_id vdisk_id vdisk_name          vdisk_UID
0  W2K8_HYPERV1_PROD 0 2 WINPRD_MIRROR_VOL01  6005076801AB00E90800000000000003
0  W2K8_HYPERV1_PROD 1 3 WINPRD_MIRROR_VOL02  6005076801AB00E90800000000000004
0  W2K8_HYPERV1_PROD 2 14 Exchange_VOL01       6005076801AB00E90800000000000014
2  VMSERVER4         0 8 DB2VOL01             6005076801AB00E90800000000000009
2  VMSERVER4         1 9 DB2VOL02_Master      6005076801AB00E9080000000000000A
2  VMSERVER4         2 10 DB2VOL03             6005076801AB00E9080000000000000B
2  VMSERVER4         3 38 Test_TP_Volume       6005076801AB00E9080000000000003A
2  VMSERVER4         4 39 Test_TP_Volume_Test  6005076801AB00E9080000000000003B
2  VMSERVER4         5 40 Test_TP_Volume_Test2 6005076801AB00E9080000000000003C
2  VMSERVER4         6 41 Test_TP_Volume_Test3 6005076801AB00E9080000000000003D
[root@Demo ~]#

[root@Demo ~]# ssh itsomon@10.18.228.105 lshostvdiskmap|grep W2K8_HYPERV1_PROD
0  W2K8_HYPERV1_PROD 0 2 WINPRD_MIRROR_VOL01  6005076801AB00E90800000000000003
0  W2K8_HYPERV1_PROD 1 3 WINPRD_MIRROR_VOL02  6005076801AB00E90800000000000004
0  W2K8_HYPERV1_PROD 2 14 Exchange_VOL01       6005076801AB00E90800000000000014
[root@Demo ~]#

[root@DemoLG-C-206-RHEL55 ~]#  ssh itsomon@10.18.228.105 lsvdisk| awk '{print
$1,$2,$5,$8}'|grep 'Exchange_VOL01\|name'
id name status capacity
14 Exchange_VOL01 online 10.00GB
[root@DemoLG-C-206-RHEL55 ~]#
```

You can customize these commands and filters to obtain an output that is most useful for you.

10.2.2 Persistent connections

A *persistent connection* is a connection that exists beyond the submission and execution of a single command. As outlined previously, the CIMOM provides a persistent connection, but it does not provide direct access to the command line. To provide a persistent connection to the command line, you must use multiple processes.

There are as many ways to provide a persistent connection to the command line as there are programming languages. Example 10-14 on page 405 shows a method of creating a persistent connection using the Korn shell. Most methods involve creating a process that connects to the cluster, writing to its STDIN stream, and reading from its STDOUT and STDERR streams. The difficulty is in detecting when the output from a command ends. Example 10-14 on page 405 shows how to detect when the output from a command ends.

You can use persistent connections in a number of ways:

- On a per-script basis

  On a per-script basis, a script opens a connection that exists for the life of the script, allowing multiple commands to be submitted. The connection ends when the script ends.

- As a stand-alone script

  With a stand-alone script approach, a connection is opened and other scripts communicate with this script to submit commands to the cluster. This approach allows the connection to be shared by multiple scripts. This in turn allows a greater number of independent scripts to access the cluster without using up all of the connection slots.

10.3 SVC command-line scripting

When connected to the cluster command line you can use small amounts of automation for various purposes, including for the following tasks:

- Repeatedly submitting a single command to a set of SVC objects
- Searching the configuration for objects conforming to certain criteria

The SVC command line is actually a highly restricted bash shell. You cannot access UNIX commands, such as **cd** or **ls**. The only commands that are available are commands that are built in, such as **echo** or **read**. In addition, redirecting inputs and outputs is not supported, but you can pipe commands together.

Example 10-9 on page 402 shows a sample script that lists all of the volumes that are not online. This script complements the **filtervalue** parameter of the **lsvdisk** command. The **filtervalue** parameter allows matches only when a property matches a value. The command-line script in Example 10-9 allows matches according to other criteria.

Example 10-9 SVC command-line script listing volumes that are not online

```
001. svcinfo lsvdisk -nohdr | while read id name IOGid IOGname status rest
002. do
003.   if  [ "$status" != "online" ]
004.   then
005.     echo "Volume '$name' \($id\) is $status"
006.   fi
007. done
```

Line 001 is of particular note. This line submits the **svcinfo lsvdisk** command and pipes the output to the **read** command, which is combined with a **while** command. This combination creates a loop that executes once per line of output from the **svcinfo** command.

The **read** command is followed by a list of variables. A line is read from the **svcinfo** command. The first word in that line is assigned to the first variable, the second word is assigned to the second variable, and so on, with any leftover words assigned to the final variable (with intervening spaces included).

In our case we use the **-nohdr** parameter, because we are not interested in this information.

Lines 003 to 006 simply check the **status** variable. If it is not equal to `online`, the information is printed to STDOUT.

10.3.1 Submitting command-line scripts

You can submit command-line scripts from an interactive prompt, if required. However, you can also submit the scripts as batch files. Example 10-10 shows how to submit scripts as batch files with **ssh**.

Example 10-10 Submission of batch file to SVC using ssh

```
root@Kanaga:/>ssh -F sampleCfg cl2 -T < batchFile.sh
MDisk 0 (mdisk0) is quorum disk 1
MDisk 1 (mdisk1) is quorum disk 2
MDisk 11 (mdisk11) is quorum disk 0
root@Kanaga:/>
```

Example 10-11 shows how to submit scripts as batch files with **plink**.

Example 10-11 Submission of batch file to SVC using plink

```
C:\scripts>"c:\Program Files\PuTTY\plink" -load itsosvccl2 -m batchFile.sh
MDisk 0 (mdisk0) is quorum disk 1
MDisk 1 (mdisk1) is quorum disk 2
MDisk 11 (mdisk11) is quorum disk 0

C:\scripts>
```

Both commands submit a simple batch file, as shown in Example 10-12 on page 403. This command lists the quorum MDisks; in a large configuration, this script can be useful.

Example 10-12 Command-line batch file batchFile.sh used in the previous examples

```
svcinfo lsmdisk -nohdr | while read id name rest
do
  svcinfo lsmdisk $id | while read key value
  do
    if [ "$key" == "quorum_index" ]
    then
      if [ "$value" != "" ]
      then
        echo "MDisk $id ($name) is quorum disk $value"
      fi
    fi
  done
done
```

10.3.2 Example SVC command-line script

Example 10-13 shows a nontrivial script that can be executed directly from the SVC command line using batch submission. The script generates a comma-separated value (CSV) table showing the number of extents that each volume gets from each MDisk.

Example 10-13 Generating CSV representation of MDisk/Volume extent usage

```
001. echo Volume,mDisk,Controller,mDisk Group,Extents;
002.
003. vdiskIds=(`svcinfo lsvdisk -nohdr | while read id rest; do echo -n "$id ";
     done`)
```

```
004. vdiskNames=(`svcinfo lsvdisk -nohdr | while read id name rest; do echo -n
      "$name "; done`)
005. vdiskNameMap=()
006. for (( i = 0 ; i < ${#vdiskNames[@]} ; i++ ))
007. do
008.   vdiskNameMap[${vdiskIds[$i]}]=${vdiskNames[$i]}
009. done
010.
011. svcinfo lsmdisk -nohdr | while read mdiskId mDiskName status mode mdgId
      mdgName capacity LUN controllerName UniqueID;
012. do
013. svcinfo lsmdiskextent -nohdr $mdiskId | while read vdiskId extents;
014.   do
015.     echo ${vdiskNameMap[$vdiskId]},$mDiskName,$controllerName,$mdgName,
        $extents;
016.   done
017. done
```

Line 001 simply generates the table header row.

Lines 003 and 004 work around a limitation of the bash shell. If you assign a value to a variable while inside a loop that is running from a pipe, that value is not available to you when outside the loop. A way to get around this situation is command substitution. Line 003 creates an array of volume IDs by looping through all of the volumes and printing the ID. This output is captured using backticks. The parentheses turn this list of IDs into an array. Line 004 performs the same process with volume names.

Lines 005 to 009 create an associative array that maps a Volume ID to a Volume name. The loop this time does not involve a pipe, so the changes to the vdiskNameMap array are available after line 009.

Line 011 begins a loop over all MDisks.

Line 013 loops over all of the MDisk Extent records for the current MDisk. A line is printed for each volume that uses extents from the current MDisk.

10.4 Server-side scripting

Server-side scripting involves scripting where the majority of the programming logic is executed on a server.

Part of server-side scripting is the generation and management of connections to the SVC cluster. Refer to 10.4.1, “Creating persistent SVC connections” on page 405 to understand how to create and manage a persistent connection to a cluster and how to manage requests coming from multiple scripts.

The Perl module handles the connection aspect of any script. Because connection management is often the most complex part of any script, it is advisable to investigate this module. Currently, this module uses transient connections to submit commands to a cluster, and it might not be the best approach if you plan to use multiple scripts submitting commands independently.

In addition to this example, scripting tools for SVC information are available at:

https://www.ibm.com/developerworks/mydeveloperworks/groups/service/html/communityview?communityUuid=5cca19c3-f039-4e00-964a-c5934226abc1

10.4.1 Creating persistent SVC connections

Example 10-14 illustrates a script that opens a persistent connection to a cluster and handles commands from other external scripts. This section describes the various features of the script to give you an idea of the issues that must be resolved.

Example 10-14 Creating a persistent connection to an SVC cluster

```
#!/bin/ksh

#
# SVC Command Queue
#
# This script implements a queue that any number of clients can write to.
# The clients provide SVC commands and the name of a file to where the output
# should be written
# Commands should be written to the queue thus:
#
#  <command><delimiter><outFile>
#
# command:   SVC command to be submitted to the cluster
# delimiter: User definable character (defined in this script)
# outFile:   File to which the output of the SVC command should be written
#
# for example: echo svcinfo lscluster -delim :+/tmp/output
#
# This would send the 'svcinfo lscluster -delim :' command to the cluster and
# place the output in the file '/tmp/output'
#
# The  '+' character delimits the command and the output file

############################################################
# Functions start
############################################################
#
# Opens an SSH pipe and purges any extraneous data from the buffer
# SSHPID gets set to the PID of the SSH process
#
# This function assumes that the connection will work
#
open_SSH_pipe()
{
    echo "Opening SSH pipe to $CL..."
    ssh -p 22 -q admin@$CL 2>&1 |&
    SSHPID=$!

    echo "Started SSH process: $SSHPID"

    #
    # Purge any data from the connection
```

```
    #
    echo "Purging connection buffer..."
    svc_transaction "svcinfo lscluster" "/dev/null"
    echo "Connection buffer purged"
}

#
# Opens the command FIFO and connects it to file descriptor 4
#
open_command_FIFO()
{
    # if the FIFO doesn't exists, create it
    if [ ! -a $FIFO ]
    then
        echo "Cannot find $FIFO so creating it"
        mkfifo $FIFO
    fi
    exec 4<$FIFO
}

#
# svc_transaction(command,outputfile)
#
# Submits the supplied command to the SVC cluster
# Send the output to the outputfile
#
svc_transaction()
{
    #
    # First, submit the command to the cluster
    # followed by a command to print the return code
    # and then a command to print the End Of Output string
    #
    print -p "$1"
    print -p "echo \$?"
    print -p "echo $COMMAND_ENDER"

    #
    # Now, read back from the SVC cluster
    #
    read -p prevLine
    while true
    do
        read -p currLine
        if [ "${currLine}" == "$COMMAND_ENDER" ]
        then
            #
            # Append the SVC Return Code to the output file
            #
            echo $RC_PREPEND$prevLine$RC_APPEND >> $2
            echo $COMMAND_ENDER >> $2
            return
        else
            echo "${prevLine}" >> $2
            prevLine=$currLine
```

```
        fi
    done
}

##########################################################
# Functions end
##########################################################


#
# CL:    cluster name - to be provided on the command line
# FIFO:  FIFO name
# DELIM: Delimiter separating commands from their output files
# COMMAND_ENDER: String that signifies that the SVC command output has ended
# RC_PREPEND:
#           : Strings that wrap round the return code from the SVC cluster
# RC_APPEND :
#
if [ "$1" == "" ]
then
    echo "Please provide a cluster name"
    return 0
fi
CL=$1
FIFO=/tmp/FIFO.$CL
DELIM=+
COMMAND_ENDER="+++COMPLETE+++"
RC_PREPEND="+++RC:"
RC_APPEND="+++"


open_SSH_pipe
open_command_FIFO

echo "Queue is ready..."
while true
do
    while read CMD <&4
    do
        OUTFILE=`echo $CMD | cut -d $DELIM -f 2`
        echo "Outfile: $OUTFILE"
        SVCCMD=`echo $CMD | cut -d $DELIM -f 1`
        echo "SVC Command: $SVCCMD"
        #
        # Probably a good idea to check that the SSH process is still running and
        # handle it if it is not.
        #
        ps -p $SSHPID | grep -q ssh
        if [ $? = 0 ]
        then
            svc_transaction "$SVCCMD" $OUTFILE
        else
            #
            # Handle the fact that SSH is no longer running
            #
```

```
153.             echo "SSH pipe seems to have died"
154.             return 0
155.         fi
156.     done
157. done
158.
159. #
160. # Closes FIFO, although we will never get here.
161. #
162. exec 4<&-
```

The open_SSH_pipe function

This function runs from lines 033 to 047 in Example 10-14 on page 405. It opens the connection to the SVC cluster. The only input is a global variable `$CL`, which holds the name of the cluster.

The function is quite simple, consisting of only three functional lines (036, 037, and 045). The first line opens the connection, redirects STDERR to STDOUT, and makes this connection a *coprocess*. A coprocess is a Korn shell term that means a process running in the background of the main process. STDIN and STDOUT of a coprocess are available to the main process for reading and writing.

Line 037 saves the process ID in `$SSHPID` for future reference.

Line 045 runs a subroutine called `svc_transaction` to clear the I/O buffer between the main process and the coprocess. This subroutine is described in "The svc_transaction function" on page 408.

The open_command_FIFO function

This function runs from lines 052 to 061 in Example 10-14 on page 405. It opens a first-in-first-out (FIFO) file. Note that a FIFO file is a UNIX construct. It is essentially a named pipe.

The FIFO file acts as a queue of commands that this script must handle. External scripts add commands to this queue, and this script takes them off the queue in the order in which they came in, and submits them to the cluster.

Lines 055 to 059 determine whether the FIFO file has been created (and create it, if it has not been created).

Line 060 opens the FIFO file with file descriptor 4 for reading. A file descriptor is simply a handle to this file; UNIX shells can read and write from these file descriptors.

The svc_transaction function

This function runs from lines 069 to 100. It submits a command to the SVC cluster, writing the output to a file. It takes two parameters. The first parameter is the full command to submit to the cluster. The second parameter is a local file, to which the output of the command must be written.

Lines 076 to 078 send information to the coprocess, that is, the persistent connection, which is done with the **`print -p`** command. First, the command is sent. Then, a request to echo the return code of the command is sent. Finally, a request to echo a known string is sent. The request to echo a known string is used to detect the end of output from the submitted command.

Lines 083 to 099 handle the gathering of the cluster's response. The command **read -p** reads from the coprocess and puts the information into the supplied variable. The loop is fairly simple:

1. Read one line from the coprocess and save as *prevLine*.
2. Start loop.
3. Read next line from coprocess as *currLine*.
4. If *currLine* matches the end of command string:
   a. Write *prevLine* to the output file with a wrapper indicating that it is the return code.
   b. Write the end of command string to the output file.
   c. End subroutine.
5. If not:
   a. Write *prevLine* to output file.
   b. Save *currLine* as *prevLine*.
6. Go to step 2.

When this subroutine is complete, the I/O buffer to the coprocess is clear.

Main script

The main script runs from line 116 to 162 and sets up the FIFO and persistent connection. The script then sits, waiting for commands to be added to the queue before submitting them to the cluster.

Lines 116 to 126 set up various variables required for execution. A more mature implementation allows these variables to be set with a configuration file, but this method suffices for a demonstration. The script requires that a cluster name is provided as a command-line parameter.

Lines 129 and 130 open the connection to the SVC cluster and a connection to the command queue (the FIFO file).

Lines 133 to 157 are the main loop. This loop monitors the command queue and handles incoming commands. This process is followed:

1. Read a line from the FIFO. This action is a blocking call that waits until a line is available.
2. Split this line on a predefined delimiter. The first field is the command destined for the cluster. The second field is the file to which all output must be directed.
3. Check that the connection to the cluster is still running. In this example, the script ends if the connection ends. A more sophisticated response re-attempts connection.
4. If the connection is still running, the command is passed to the **svc_transaction** subroutine.

The final line is never reached due to the constant loop, but its action is to close the file descriptor to the FIFO queue.

Command submission

If multiple scripts are trying to write to the command queue that was created previously, actions must be taken to ensure that the writes are *atomic*. Atomic means if two scripts attempt to write to the command queue at overlapping times, the commands written do not interleave.

Example 10-15 shows a simple C program that performs an atomic append to a file. This C program is a perfect utility for writing to the FIFO file when multiple scripts are accessing the file at the same time. It blocks on the **write** function on line 059 until it is free to write. All other scripts attempting to write are blocked at the same time.

Example 10-15 Utility that performs atomic appends to file

```
001. #include <stdlib.h>
002. #include <stdio.h>
003. #include <fcntl.h>
004. #include <string.h>
005. #include <unistd.h>
006.
007. int main(int argc, char *argv[])
008. {
009.     /*
010.      * First parameter is the file
011.      * Second parameter is the text
012.      */
013.
014.      char *filename; // Name of file to append to
015.      int fileD;      // File descriptor (for file access)
016.      char *text;     // Text to append to the file in question
017.      int textLen;    // Length of the text that is appended to the file
018.
019.     /*
020.      * Ensure that we have the required parameters
021.      */
022.     if (argc != 3)
023.     {
024.         printf("Usage: append <file> <text>\n");
025.         return -1;
026.     }
027.
028.     filename = argv[1];
029.
030.     /*
031.      * The text appended to the file will have a "newline" appended to it
032.      * We also need to include space in the buffer for the null-terminator
033.      */
034.     textLen = strlen(argv[2]) + 2;
035.     text = (char *) malloc((textLen)*sizeof(char));
036.     if(text == NULL)
037.     {
038.         printf("Could not allocate memory\n");
039.         return -2;
040.     }
041.     strcpy(text,argv[2]);
042.     strcat(text,"\n");
043.
044.     /*
045.      * Open the file that we are appending to
046.      * NB: We do not attempt to create this file if it does not exist
047.      */
048.     fileD = open(filename, O_WRONLY | O_APPEND, 0);
049.
```

```
    if(fileD < 0)
    {
        perror("Cannot open append file for writing");
        return -2;
    }

    /*
     * Write to append file and then close the file
     */
    write(fileD,text,textLen);

    close(fileD);

    return 0;
}
```

10.4.2 Handling the SVC response

From this point forward, we assume that you have an implementation that enables you to submit commands to the cluster and retrieve the response in an effective manner. The next step is to handle the responses usefully.

Handling errors

The publication *IBM System Storage SAN Volume Controller and IBM Storwize V7000, Version 6.3.0: Command-Line Interface User's Guide*, GC27-2287, provides a list of the possible errors that can result from issuing an SVC command. It is important to know how to recognize and handle these command errors.

Every SVC cluster error has a unique code of the following format:

- `CMMVC`*`xxxx`*`E` for an error
- `CMMVC`*`xxxx`*`W` for a warning

These codes map to a single outcome, although different commands can result in identical errors. For example, the `CMMVC5786E` code maps to the `The action failed because the cluster is not in a stable state` error, which can be returned for any command that seeks to change the cluster configuration by adding or removing an object.

It is important that any automation solution, at a minimum, reports these errors accurately, because the errors are needed for troubleshooting. At the next level, an automation solution must be able to handle errors automatically. A large proportion of the errors refer to invalid requests, such as trying to use objects that do not exist, or trying to change the state of an object to an invalid state. You can avoid these errors by ensuring that your scripts check object states before changing them. You can also use these errors as checks; if you encounter one of these errors, something is obviously wrong, and the user must be alerted.

The svcinfo commands

These commands are critical to automation, because they indicate the current state of the cluster. It is important to ascertain object states before changing them if you intend to implement an enterprise-level automation solution.

If you use the SVCTools Perl module, you can access cluster object properties as a result of executing the functions that this module provides. This section describes how to access these properties through the Korn shell.

It is strongly advised that commands submitted to the SVC cluster by an automation solution are submitted using the **-delim** parameter. In general, we use the colon character (:) as the delimiter. The colon character as the delimiter is just one convention, and you can use any character that you do not otherwise expect in the output.

After the output has been captured, each line represents an object. This line can then be split into words, separated by the chosen delimiter. Each of these words represents a property.

At this point, you have access to the information that you need. Write your scripts to perform the processes according to your business logic.

Be aware that various commands are available on the IBM SAN Volume Controller only, or on IBM Storwize V7000 only, and not necessarily on both anymore. Table 10-1 lists some examples.

Table 10-1 Command availability

Command	IBM SAN Volume Controller	IBM Storwize V7000
lsnode	Available	Not available
svcinfo lsnode	Available	Available
lsnodecanister	Not available	Available

The svctask commands

When these commands succeed, the feedback is minimal. The **svctask** commands that create objects return the ID of the created object. The format of the response is `<object type>, id [<new object's id>], successfully created`. Thus, the newly created object's ID can be captured from this line. Example 10-16 shows how you can capture the newly created SVC object ID in a Perl program.

Example 10-16 Capturing newly created SVC object ID in Perl

```
#
# $line is a SCALAR containing the response line from SVC
#
if ($line =~ /(.*), id \[(\d*)\], successfully created/)
{
   $objectType = $1;
   $newId = $2;
}
```

The SVC object ID is important, because it enables you to execute further commands against the newly created object.

Other commands complete successfully with no feedback other than a return code of zero (0). For instance, commands that change an object return no feedback when they are successful. It might be prudent to request their details to ensure that any data structures that you have created are in sync with the cluster objects' states.

10.5 SVC CIMOM programming

Automation is discussed in 10.4, "Server-side scripting" on page 404 by connecting directly to the cluster command line. As outlined in 10.2, "Creating connections" on page 393, this method requires you to manage connections to prevent failures when the connection limit is met. In addition, it is difficult to manage connections without sacrificing the user authentication process.

CIMOM programming provides the opportunity to handle multiple connections from multiple sources while maintaining security. CIM clients connect to the CIMOM with a user name and password and then invoke methods to execute commands. *IBM System Storage SAN Volume Controller, Version 6.3.0, CIM Agent Developer's Guide,* GC27-2288, contains details of the CIM classes, associations, and methods that you need to interact with the cluster.

Creating a CIM client is straightforward when you have found a suitable framework. Our investigations have uncovered a few frameworks, but the Java Web Based Enterprise Management (WBEM) Service project seems to be one of the more widely referenced frameworks. You can obtain it at the following site:

http://wbemservices.sourceforge.net/

Implementations are available in other languages, including C++ and Python, but our experience with Java directed us to this implementation.

Example 10-17 shows a simple Java program that connects to an SVC CIMOM. Connecting is the first step in automating with the CIMOM.

Example 10-17 Java program for connecting to SVC CIMOM

```
import java.util.*;

import javax.wbem.cim.*;
import javax.wbem.client.*;

public class ITSOClient {

   public static void main(String[] args)
   {
      String username = args[0];
      String password = args[1];
      String masterConsoleIP = args[2];
      String masterConsoleSecurePort = args[3];
      UserPrincipal user = new UserPrincipal(username);
      PasswordCredential pwd = new PasswordCredential(password);
      CIMNameSpace ns = new CIMNameSpace("https://"+
            masterConsoleIP+":"+
            masterConsoleSecurePort+"/root/ibm");

      CIMClient client = null;
      try
      {
         System.out.println("Connecting to CIMOM");
         client = new CIMClient(ns,user,pwd);
      }
      catch (CIMException e)
      {
         // Handle the CIM Exception
         e.printStackTrace();
      }
}
```

When connected to the CIMOM, the next step is to identify the cluster with which you want to work. Before moving on to that topic, however, it is useful to briefly examine CIM agent concepts.

10.5.1 CIM concepts

A full definition of the CIM specification is beyond the scope of this book, but a description of a few concepts here will enable you to experiment with this technology. This section assumes familiarity with object-oriented programming (OOP) concepts:

Namespace The scope within which all of the CIM classes are defined. For SVC, the namespace contains all the SVC CIM classes that your scripting will manipulate.

Class The definition of an object within the SVC hierarchy. It is directly equivalent to an OOP class.

Instance An instance of an object that is a member of a CIM class. It is directly equivalent to an OOP object and contains all the properties of an object in the SVC configuration.

Method A way to implement a function on a class.

Property An attribute that is used to characterize the instance of a class.

Qualifier A value that defines the behavior of a property, classes, and methods.

Reference A pointer to a CIM instance, which is a *typed* string that is identified as a reference to the instance in question.

Object path A formatted string that is used to access the name spaces, classes, and instances.

Association A class with two references that defines a relationship between the two referenced classes.

The CIM classes are part of a broader, cross-vendor specification and thus do not always map exactly onto SVC object types. *IBM System Storage SAN Volume Controller CIM Agent Developer's Reference,* SC26-7904, shows all of the relevant classes.

10.5.2 SVC concepts mapping to CIM concepts

To administer copy services through the CIMOM you need to understand how to translate between SVC and CIM concepts. Table 10-2 shows their relationships.

Table 10-2 Relating SVC concepts to CIM concepts

SAN volume Controller concept	CIM	
	CIM name	CIM concept
Cluster	IBMTSSVC_Cluster	Class
ClusterName	ElementName	Property
Cluster ID	Name	Property
VDisk	IBMTSSVC_StorageVolume	Class
VDisk ID	DeviceID	Property

SAN volume Controller concept	CIM	
	CIM name	CIM concept
FlashCopy mapping	IBMTSSVC_LocalStorageSynchronized	Association
FlashCopy mapping status	SyncState	Property
mkfcmap	AttachReplica	Method
preparefcmap	ModifySynchronization	Method
startfcmap	ModifySynchronization	Method
Remote Copy relationship	IBMTSSVC_RemoteStorageSynchronized	Association
Remote Copy relationship state	NativeState	Property
mkrcrelationship	AttachReplica	Method
startrcrelationship	ModifySynchronization	Method

10.5.3 Creating and starting a FlashCopy mapping

Example 10-18 on page 415 is the main method from a Java class that is designed to create and start a FlashCopy mapping. The other methods that are called are described in later examples.

This example demonstrates how CIMOM methods can control the cluster. In this section, the term *method* refers to a Java method. The term *Method* (with an initial capital) refers to a CIM Method.

Example 10-18 Java main method for creating and starting FlashCopy mapping

```
/*
 * FC Mapping states
 */
private static UnsignedInt16 INITIALIZED = new UnsignedInt16(2);
private static UnsignedInt16 PREPARING   = new UnsignedInt16(3);
private static UnsignedInt16 PREPARED    = new UnsignedInt16(4);

public static void main(String[] args) throws CIMException
{
   /*
    * First step is to connect to the CIMOM
    */
   UserPrincipal user = new UserPrincipal("superuser");
   PasswordCredential pwd = new PasswordCredential("itso13sj");
   CIMNameSpace ns = new CIMNameSpace("https://9.43.86.115:5989/root/ibm");

   CIMClient client = null;

   client = new CIMClient(ns,user,pwd);

   /*
    * Next, select the cluster that we're interested in
    */
   CIMInstance chosenCluster = getCluster("ITSOCL1",client);
```

```
/*
 * At this point, the relevant cluster has been selected
 * and 'chosenCluster' is a CIMInstance of this cluster
 *
 * Get the Config Service of this cluster
 */
CIMObjectPath cService = getConfigService(chosenCluster, client);

/*
 * Now, get all of the Volumes in this cluster
 */
Map<Integer,CIMObjectPath> vdisksById = getVDisks(chosenCluster,client);

/*
 * Select the FlashCopy Source
 *
 * In this case, Volume 10 is our source
 * Volume 11 is our target
 */
CIMObjectPath fcSrc = vdisksById.get(new Integer(10));
CIMObjectPath fcTgt = vdisksById.get(new Integer(11));

/*
 * Now, create FC Mapping
 */
CIMValue rc = makeFlashCopyMapping("CIMOMTestMap", fcSrc, fcTgt, cService,
   client,false);

/*
 * Now that this has been created, we need to get an
 * Object Path to the newly created Association
 */
List<CIMObjectPath> fcMaps = getFCMappings(fcSrc, client);
CIMObjectPath fcMapping = fcMaps.get(0);

/*
 * Now, we prepare the FC Mapping
 */
CIMArgument[] outArgs = new CIMArgument[2];
rc = prepareFCMapping(cService, fcMapping, client, outArgs);
System.out.println("Got value:"+
   Integer.toHexString(Integer.parseInt(rc.toString())));

/*
 * Loop until it is prepared
 */
CIMValue fcMapState = new CIMValue(PREPARING);
while(fcMapState.equals(new CIMValue(PREPARING)))
{
   CIMInstance fcMapInfo = client.getInstance(fcMapping);
   fcMapState = fcMapInfo.getProperty("SyncState").getValue();
}

/*
 * Now start the FC Mapping
```

```
    */
   rc = startFCMapping(cService, fcMapping, client, outArgs);
   System.out.println("Got value:"+
      Integer.toHexString(Integer.parseInt(rc.toString())));
}
```

The assumption in this example is that your Java program is designed to control the same cluster every time. It is a relatively simple process to make it more flexible, but that is left to you.

The getCluster method

Example 10-19 shows the `getCluster` method. This method returns the CIM instance that corresponds to the cluster with the supplied name by enumerating all of the instances of the class IBMTSSVC_Cluster and then checking the name of each one. When an instance is found that matches the supplied name, an object path to that instance is returned.

Example 10-19 The getCluster method

```
static private CIMInstance getCluster(String clusterName, CIMClient client) throws
CIMException
{
   CIMInstance chosenCluster = null;
   Enumeration<CIMInstance> clusters =
      client.enumerateInstances(new CIMObjectPath("/root/ibm:IBMTSSVC_Cluster"));

   while(clusters.hasMoreElements())
   {
      CIMInstance possibleCluster = clusters.nextElement();
      String possibleName =
         possibleCluster.getProperty("ElementName").getValue().toString();

      if(possibleName.equals("\""+clusterName+"\""))
      {
         chosenCluster = possibleCluster;
      }
   }

   return chosenCluster;
}
```

The getConfigService method

Example 10-20 shows the `getConfigService` method. The CIM_StorageConfigurationService class has no direct equivalent in an SVC, but an Instance of this class is required for invoking Methods.

In this method, in our example, we request all of the instances associated with the supplied cluster. The association that connects a cluster to its configuration service is CIM_HostedService. A cluster has only a configuration service associated with it, so we can select the first object path in the enumeration.

Example 10-20 The getConfigService method

```
static private CIMObjectPath getConfigService(CIMInstance cluster, CIMClient
client) throws CIMException
{
   Enumeration<CIMObjectPath> configServices = null;
   configServices = client.associatorNames(
         cluster.getObjectPath(),
         "CIM_HostedService",
         "CIM_StorageConfigurationService",
         null,
         null);

   return configServices.nextElement();
}
```

The getVDisks method

Example 10-21 shows the `getVDisks` method, which returns a map, relating Volume IDs (as integers) to IBMTSSVC_StorageVolume object paths. The method requests all of the IBMTSSVC_StorageVolume instances that are associated with the provided cluster instance.

Example 10-21 The getVDisks method

```
static private Map<Integer,CIMObjectPath> getVDisks(CIMInstance cluster, CIMClient
client) throws CIMException
{
   Enumeration<CIMObjectPath> vdisks = client.associatorNames(
         cluster.getObjectPath(),
         null,
         "IBMTSSVC_StorageVolume",
         null,
         null);

   Map<Integer,CIMObjectPath> vdisksById = new HashMap<Integer, CIMObjectPath>();

   while(vdisks.hasMoreElements())
   {
      CIMObjectPath vdiskOP = vdisks.nextElement();
      CIMValue vdiskId = vdiskOP.getKey("DeviceID").getValue();
      String idAsString = vdiskId.toString();
      String idNoQuotes = idAsString.substring(1, idAsString.length()-1);
      vdisksById.put(Integer.parseInt(idNoQuotes), vdiskOP);
   }

   return vdisksById;
}
```

The makeFlashCopyMapping Method

Example 10-22 shows the `makeFlashCopyMapping` method. This method invokes `AttachReplica` against the cluster configuration service.

CIM Methods take typed parameters. In this method, you can see the use of the `argRef`, `argString`, and `argUint16` methods, which are defined later in this section. For now, you need to know that they act as shortcuts to generating the required arguments for the CIM Method.

Note the use of `CopyType`. The `AttachReplica` method is used for FlashCopy, Metro Mirror, and Global Mirror, and the `CopyType` argument indicates which type is required.

Example 10-22 The makeFlashCopyMapping Method

```
static private CIMValue makeFlashCopyMapping(
            String name,
            CIMObjectPath source,
            CIMObjectPath target,
            CIMObjectPath configService,
            CIMClient client,
            boolean autodelete) throws CIMException
{
   CIMArgument src = argRef("SourceElement", source, "IBMTSSVC_StorageVolume");
   CIMArgument tgt = argRef("TargetElement", target, "IBMTSSVC_StorageVolume");
   CIMArgument fcName   = argString("ElementName",name);
   CIMArgument type   = argUint16("CopyType",autodelete?5:4);

   CIMArgument[] inArgs  = {src,tgt,fcName,type};
   CIMArgument[] outArgs = new CIMArgument[1];

   CIMValue rc = client.invokeMethod(configService,
         "AttachReplica",
         inArgs,
         outArgs);
   return rc;
}
```

The getFCMappings method

Example 10-23 shows the `getFCMappings` method. This method returns a list of all of the FCMappings associated with the provided Volumes. It requests a list of all of the associations that reference the provided IBMTSSVC_StorageVolume. Currently, all of the Java WBEM services methods of this type return enumerations. This method converts the enumerations to a list for ease of use.

Example 10-23 The getFCMappings method

```
static private List<CIMObjectPath> getFCMappings(CIMObjectPath vdisk, CIMClient
client) throws CIMException
{
   Enumeration<CIMObjectPath> assocs = client.referenceNames(
            vdisk,
            "IBMTSSVC_LocalStorageSynchronized",
            null);

   return Collections.list(assocs);
}
```

The prepareFCMapping method

Example 10-24 shows the `prepareFCMapping` method. This method prepares a FlashCopy mapping. Much like the `AttachReplica` Method, the `ModifySynchronization` Method is used to control FlashCopy, Metro Mirror and Global Mirror; the Operation parameter indicates what you actually want to do.

Example 10-24 The prepareFCMapping method

```
private static CIMValue prepareFCMapping(
            CIMObjectPath configService,
            CIMObjectPath fcMapping,
            CIMClient client,
            CIMArgument[] outArgs) throws CIMException
{
   CIMArgument operation = argUint16("Operation", 6);
   CIMArgument synch = argRef("Synchronization",
fcMapping,"IBMTSSVC_FlashCopyStorageSynchronized");

   CIMArgument[] inArgs  = new CIMArgument[]{operation,synch};
   outArgs = new CIMArgument[2];

   return client.invokeMethod(configService,
         "ModifySynchronization",
         inArgs,
         outArgs);
}
```

The startFCMapping method

Example 10-25 on page 420 shows the `startFCMapping` method. This method starts a FlashCopy mapping.

As in Example 10-24, this method invokes the `ModifySynchronization` Method. Other than using a different Operation parameter, this method is identical.

Example 10-25 The startFCMapping method

```
private static CIMValue startFCMapping(
      CIMObjectPath configService,
      CIMObjectPath fcMapping,
      CIMClient client,
      CIMArgument[] outArgs) throws CIMException
{
   CIMArgument operation = argUint16("Operation", 4);
   CIMArgument synch = argRef("Synchronization",
fcMapping,"IBMTSSVC_FlashCopyStorageSynchronized");

   CIMArgument[] inArgs  = new CIMArgument[]{operation,synch};
   outArgs = new CIMArgument[2];

   return client.invokeMethod(configService,
         "ModifySynchronization",
         inArgs,
         outArgs);
}
```

Argument generators

This class uses three argument generators:

- `argUint16`, which is shown in Example 10-26
- `argString`, which is shown in Example 10-27
- `argRef`, which is shown in Example 10-28 on page 422

The argUint16 method

This method returns an unsigned 16-bit integer type argument, as shown in Example 10-26.

Example 10-26 The argUint16 method

```
static private CIMArgument argUint16(String name, int arg)
{
   return new CIMArgument(
         name,
         new CIMValue(
               new UnsignedInt16(arg),
               new CIMDataType(CIMDataType.UINT16)
               )
         );
}
```

The argString method

This method returns a string type argument, as shown in Example 10-27.

Example 10-27 The argString method

```
static private CIMArgument argString(String name, String str )
{
   return new CIMArgument(
         name,
         new CIMValue(
               str,
               new CIMDataType(CIMDataType.STRING)
               )
         );
}
```

The argRef method

This method returns a reference type argument, which is a reference to the instance that is indicated by the provided object path. Refer to Example 10-28.

Example 10-28 The argRef method

```
static private CIMArgument argRef(
      String name,
      CIMObjectPath path,
      String className )
{
   return new CIMArgument(
         name,
         new CIMValue(
               path,
               new CIMDataType(className)
               )
         );
}
```

10.6 Logging

The object of automation is to remove the human factor from performing systems operations. Many steps in a process follow directly from their previous step with little or no human decision-making necessary.

When performed by a human operator, these processes have a natural “logging.” A user recalls which steps were taken and the outcome of each step. When the human operator is removed from a process, it is important to log each step to aid in auditing and troubleshooting.

The SVC audit log registers whenever an **svctask** command is successfully executed. The SVC error log also registers when configuration events complete. These logs are useful, but it is important to generate your own logging as part of your automation solution.

You must log the following actions with a time and date stamp:

- Submission of commands to the SVC, including *full* SVC command-line interface (CLI) commands or CIM method invocations
- Completion of commands sent to the SVC (return codes and messages)
- IDs of generated SVC objects
- Full explanation of why an automation solution did not complete

To tally server, SVC, and automation logs, it is important that all systems have a correctly set clock. The SVC cluster supports the Network Time Protocol (NTP) protocol now. Use

Settings → **General** from the GUI menu to setup the NTP source as shown in Figure 10-12 on page 423.

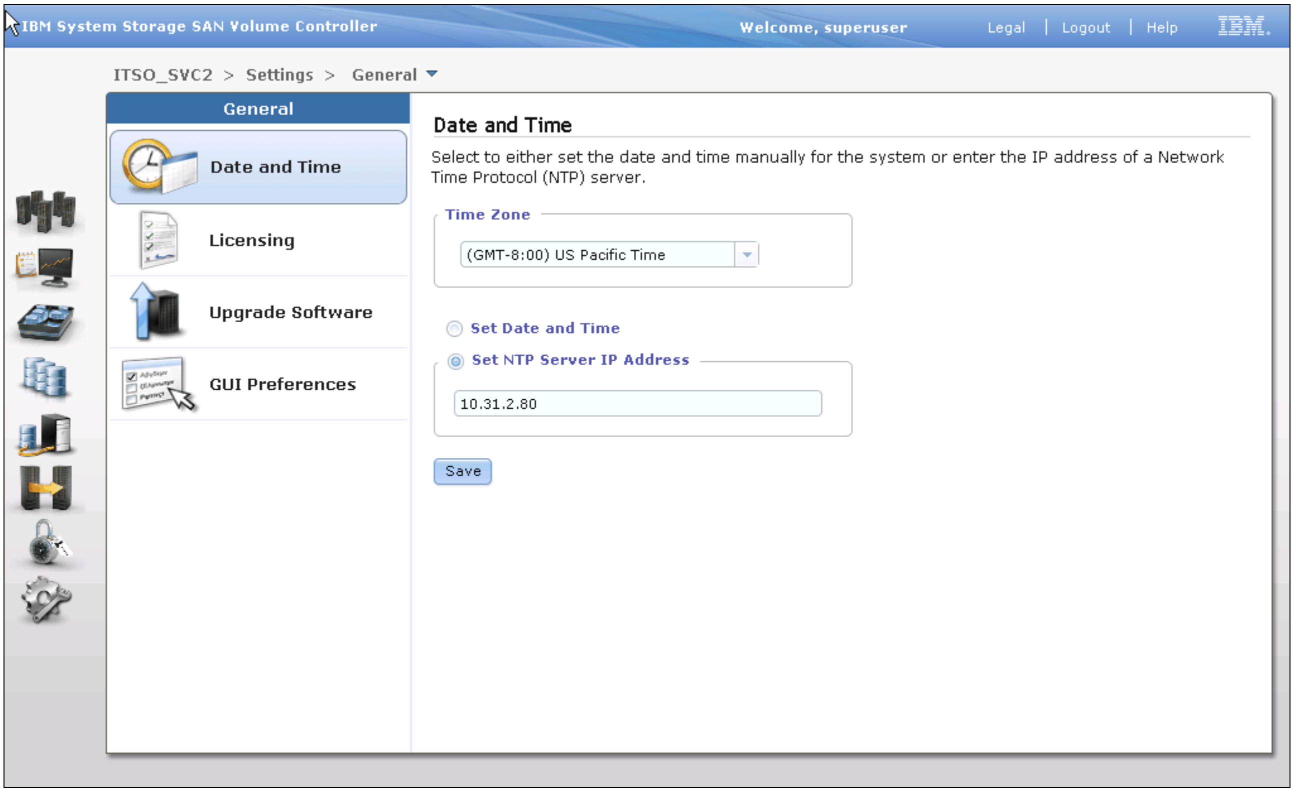


Figure 10-12 NTP server setting

A useful script example is to synchronize the time with an NTP server through proxy. On a UNIX server with access to a cluster, you can use the commands that are shown in Example 10-29 to synchronize the SVC clock with your local clock.

This script requires user interaction because the local server's time zone might map to multiple possible cluster time zones. This script can be altered to be fully automatic by hard-coding the time zone code.

Example 10-29 Setting SVC cluster clock to local time

```
!/usr/bin/ksh

locTZ=`date +%Z`
sshInvocation="ssh -F sampleCfg cl2"
possTZs=`$sshInvocation svcinfo lstimezones -delim : | grep $locTZ`
echo "Please select a timezone"
for TZ in $possTZs
do
  echo $TZ
done
echo "Selection (numerical code): \c"
read choice
```

```
echo You chose: $choice

$sshInvocation svctask settimezone -timezone $choice
$sshInvocation svctask setclustertime -time `date +%m%d%H%M%Y`
```

10.7 Auditing

When commands are successfully executed on the SVC cluster, an entry is made in the SVC audit log. Example 10-31 shows a sample audit log entry. The audit log steadily fills up with entries until it is dumped manually with the **svctask dumpauditlog** command, or until it gets too full and dumps automatically. The entries are dumped to an audit log dump file. Example 10-30 shows the retrieval of that file.

Example 10-30 Flushing and retrieving the audit log

```
root@Kanaga:/>ssh -F sampleCfg cl2 svctask dumpauditlog

root@Kanaga:/>ssh -F sampleCfg cl2 svcinfo lsauditlogdumps
id              auditlog_filename
0               auditlog_0_1410_20080815123751_0000020066403a44

root@Kanaga:/>scp -F sampleCfg cl2:/dumps/audit/auditlog_0_1410_20080815123751_000
0020066403a44 .
auditlog_0_1410_20080815123751_0000020066403a44                   100%  376KB
376.0KB/s   00:00
```

Example 10-31 shows the audit log entry.

Example 10-31 Audit log entry

```
Auditlog Entry 1410
        Sequence Num    : 1410
        Timestamp       : Fri Aug 15 12:37:51 2008
                        : Epoch + 1218803871
        Cluster User    : admin
        SSH Label       : icat
        SSH IP Address  : 9.43.86.115
        ICAT User       : superuser
        Result Code     : 0
        Result Obj ID   :
        Action Cmd      : svctask detectmdisk
```

The elements in Example 10-31 are:

- `Auditlog Entry number`

  This number indicates the position of the entry in the audit log dump file. It is unique within that file.

- `Sequence Num`

  This number is a unique number that continually increments. If the entry given in Example 10-31 is the last entry in a dump file, the first entry in the next dump file has an *`Auditlog Entry number`* of 1 and a *`Sequence Num`* of 1411.

- `Timestamp`

  This time is the time when the command was run. *Epoch* refers to a standard method of measuring time in a UNIX environment; an epoch is the number of seconds since midnight Coordinated Universal Time (UTC) of 1 January 1970, not counting leap seconds.

- `Cluster user`

  The user name, which is used to access the cluster CLI, is either `service` or `admin`. The user name exists, even when the GUI or CIMOM is used because, ultimately, the CIMOM accesses the cluster through the CLI.

- `SSH Label`

  The SSH Label is the ID given to the public key that authenticated the connection. It is a useful way to track who submitted a command. If private keys are truly kept private, this ID can be an effective audit policy.

- `ICAT User`

  This element is populated when the connection to the cluster is made through the CIMOM. The Interactive Code Analysis Tool (ICAT) user name that is used to connect to the CIMOM is the one that is logged.

- `Result Code`

  The Result Code is either `0` or `1`, depending on whether the command is successful. If the command is an **mk*xxx*** command, then `Result Obj ID` holds the ID of the object that was created.

- `Action Cmd`

  This element lists the actual command that was submitted.

This information is sufficient to audit exactly who submitted which commands and when. The audit log is only as useful as the policies that are put in place to control access. If private keys and user names are closely controlled, the audit log gives a precise history of who contacted the cluster.

11

Software solutions and services based on IBM SVC Replication Family Services

Many software solutions and services are based on IBM SVC Replication Family Services. These solutions and services are designed to provide simplified management in the following ways:

- Automating administration and configuration of these services with wizard-based session and copy set definitions
- Providing simple operational control of copy services tasks, including starting, suspending and resuming in the same direction or opposite directions
- Disaster recovery (DR) planning and execution with integrated storage and replication functions
- Application integration for application-consistent copies, backup and restore
- Offering tools for monitoring and managing copy sessions

This chapter discusses these solutions and services and explains how your enterprise can benefit from their use. The following topics are covered:

- IBM Tivoli Storage Productivity Center for Replication (TPC-R) with IBM SVC Replication Family Services
- IBM Tivoli Storage FlashCopy Manager
- IBM SVC plug-ins
- IBM SVC/Storwize V7000 disaster recovery solution for VMware environments
- IBM SVC/Storwize 7000 FlashCopy and Symantec NetBackup
- IBM SVC/Storwize V7000 Remote Copy and Veritas Storage Foundation Enterprise HA

11.1 IBM Tivoli Storage Productivity Center for Replication with IBM SVC Replication Family Services

IBM Tivoli Storage Productivity Center for Replication (TPC-R) is powerful software that you can use to manage your IBM SVC Replication Family Services. It provides the following features and benefits:

- It provides a web-based user interface that is designed to ease management tasks for Replication Family Services and improve administrator productivity. It offers high performance and simple navigation to help improve ease of use.
- It installs quickly and deploys easily. With two servers, one server can run in active standby and be ready to take over if the active server fails.
- It provides 2-way site switch capability and is designed for quick implementation to get your replication solution up and running quickly. It is designed for complex DR solutions and has local or remote capabilities.
- It offers a DR Planner so you can plan for replication when storage is provisioned, and provides improved system availability.
- It provides Multiple Global Mirror support including enhanced monitoring capabilities and the ability to create more GM sessions, thereby improving performance.

TPC-R can be a valuable addition to your management tools. This section explains how to use TPC-R Version 4.2.2.1, the latest version at the time of writing.

Although providing an exhaustive reference to TPC-R with IBM SAN Volume Controller (SVC) is beyond the scope of this book, you are introduced to its basic functions here so you can determine whether this package is appropriate in your environment.

For more detailed information about TPC-R, refer to the IBM Tivoli Storage Productivity Center Information Center:

http://publib.boulder.ibm.com/infocenter/tivihelp/v4r1/index.jsp?topic=%2Fcom.ibm.help.ic.doc%2Fusing_system%2Fabout_the_system.html

You can also refer to the IBM Redbooks publications about TPC-R, which are available at:

http://www.redbooks.ibm.com/cgi-bin/searchsite.cgi?query=tpc+AND+replication

Most of the information in *IBM TotalStorage Productivity Center for Replication Using DS8000*, SG24-7596, also applies to the SVC. The greatest difference between Copy Services for the DS8000 and the SVC is that Global Mirror with the DS8000 involves “Journal Volumes” that are not used with the SVC. In addition, three-site replication is not available with the SVC.

TPC-R and IBM SVC demo with a possible recovery scenario you can see at:

http://www.youtube.com/watch?v=09IyY6DVqe4

11.1.1 TPC-R compared to SVC/SSPC console

So you might wonder why to use TPC-R instead of the native abilities of the SVC console. If you already use TPC-R for the DS8000, IBM DS6000™, or IBM TotalStorage Enterprise Storage Server® (ESS), deciding to use it for the SVC is a relatively simple decision; there is no reason to not use it. Otherwise, you must carefully examine the features and capabilities of

TPC-R and weigh them against the SVC/System Storage Productivity Center (SSPC) console.

TPC-R includes the following helpful features:

- A single TPC-R server can handle Copy Services for many SVC Clusters (or DS8000 or DS6000 units) using the same console. It is designed to accommodate even the largest Copy Services installations, consisting of up to thousands of Consistency Groups (called “Sessions” in the software).
- Site Awareness: If you use Metro or Global Mirror, TPC-R is aware of which unit is installed at which site, and will issue mirroring commands through the remote site if the primary site has failed.
- Server failover: If the TPC-R server has failed, operations can be failed over to another TPC-R server if you have a secondary site and secondary TPC-R server. This function is quite useful in disaster recovery scenarios.
- The TPC-R GUI can be somewhat easier to use than the SVC/SSPC console.

However, using TPC-R can be less useful in the following circumstances:

- For large Copy Services installations, TPC-R can require substantial server resources.
- For failed operations, the logging messages might be somewhat less useful than if executing SVC commands directly.

11.1.2 Setting up TPC-R for use with the SVC

This section assumes that you have already installed TPC-R on the server of your choice and have logged into it with your web browser. Your console might look similar to Figure 11-1 on page 430.

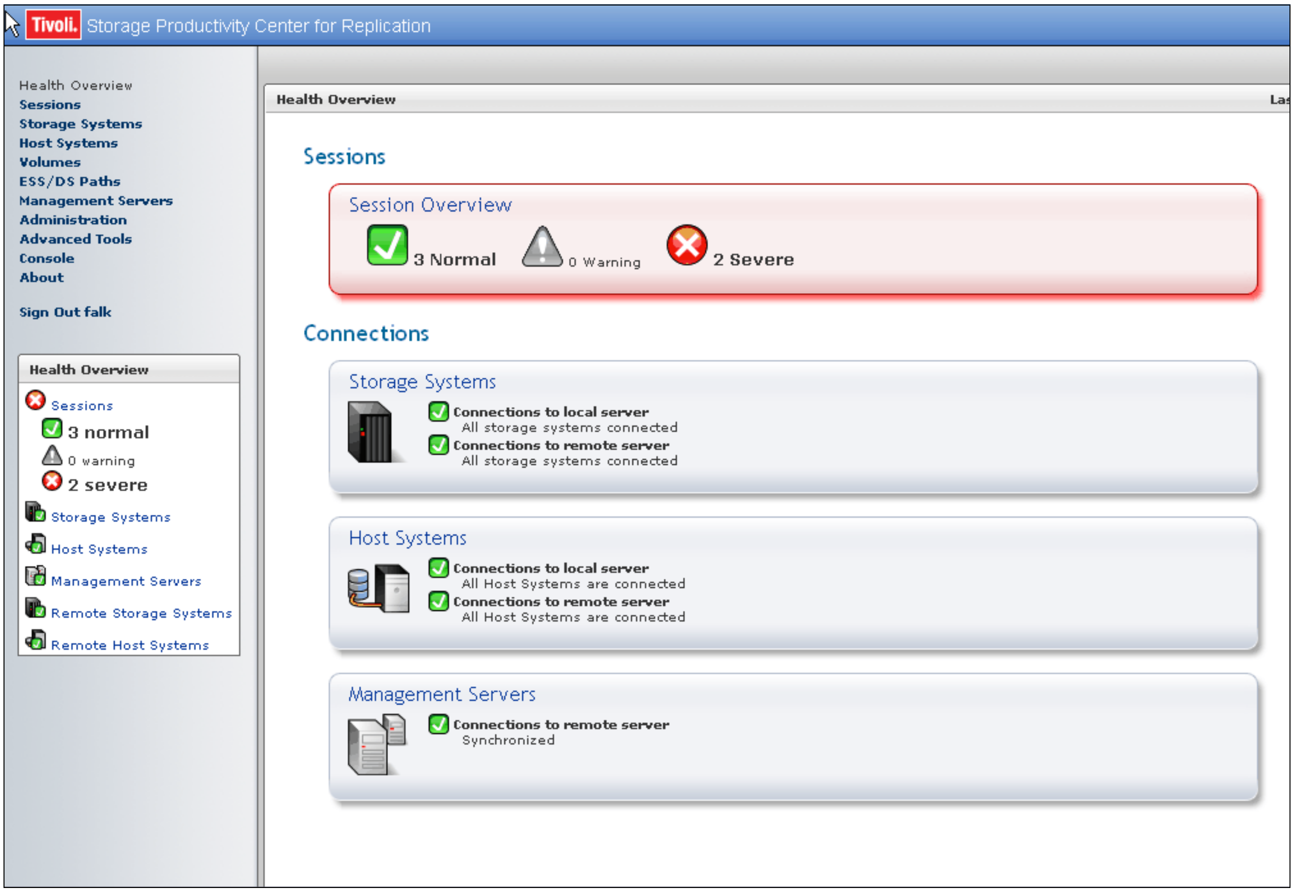


Figure 11-1 Starting the TPC-R Console

Follow these steps to start TPC-R:

1. Select **Storage Systems** → **Add Storage Connection**. The Add Storage System window is shown in Figure 11-2 on page 431 is displayed.

   In this case, **SAN Volume Controller / Storwize V7000 / Storwize V7000 Unified (Direct Connection)** were selected. After TPC-R connects directly to the storage, press **Next**.

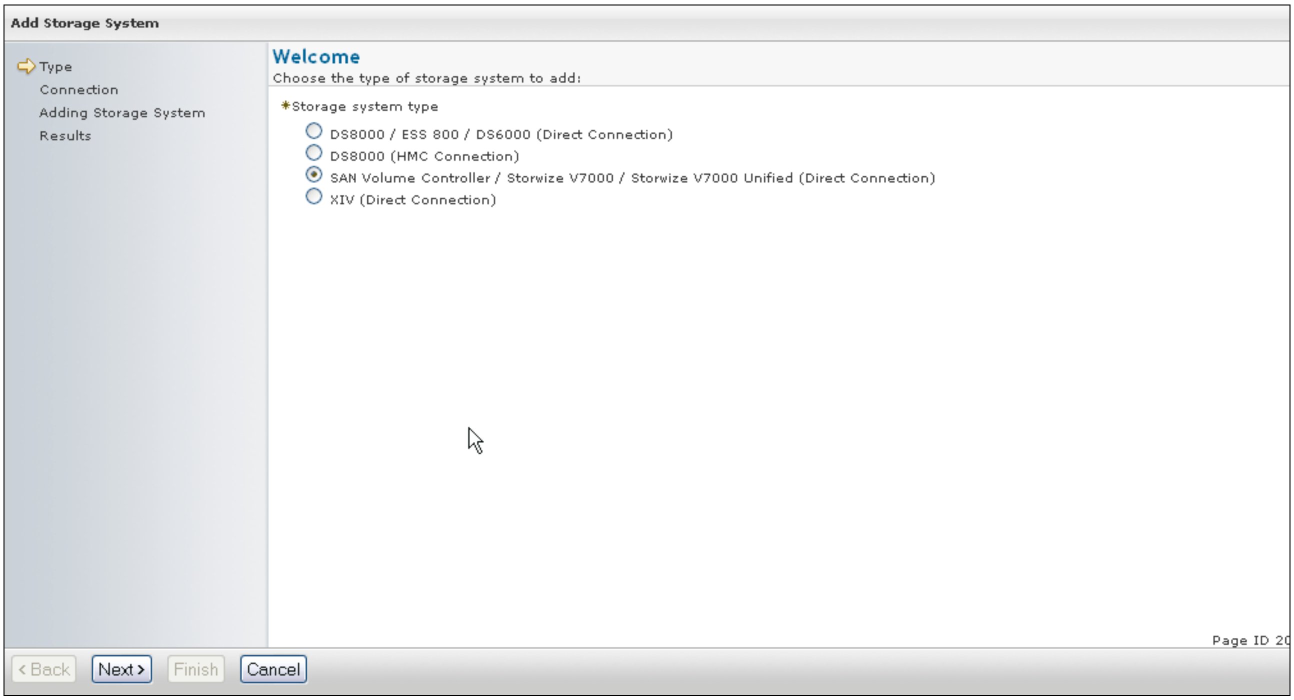


Figure 11-2 Add Storage System window

2. On the following window, fill in the details of the SVC/Storwize V7000 as shown in Figure 11-3.

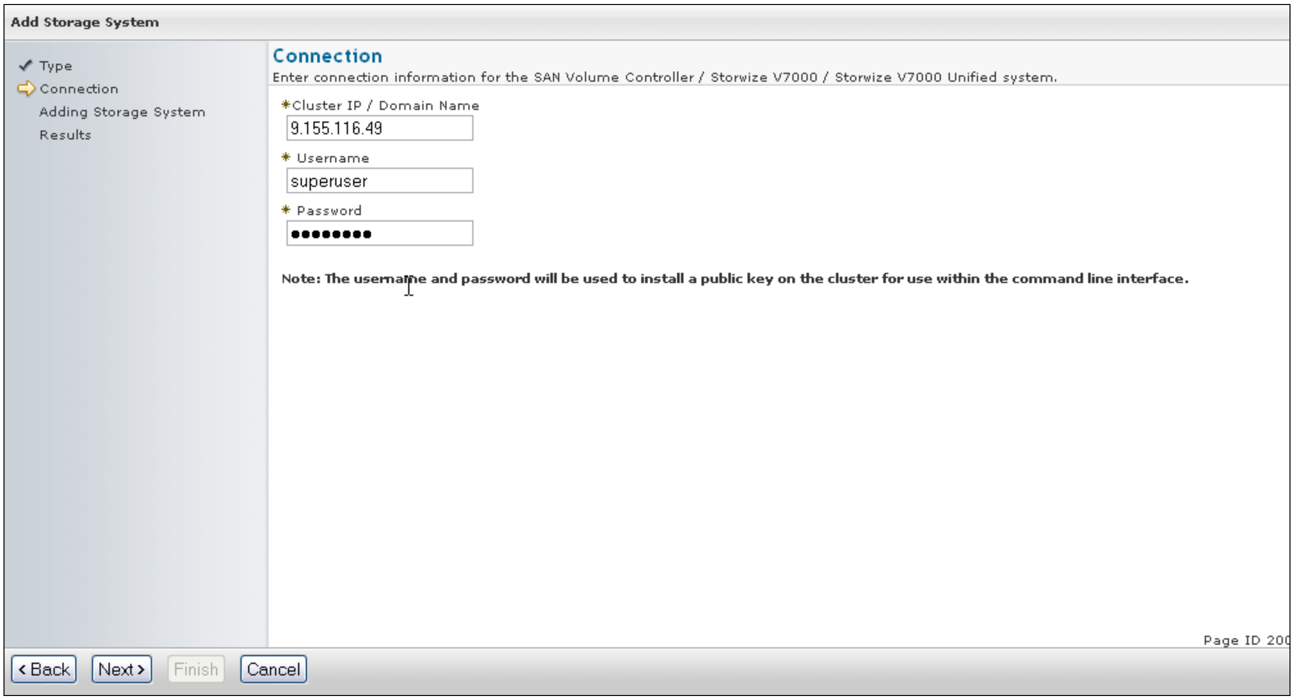


Figure 11-3 SVC Connection window

Next, the wizard displays the result and indicates that the system was successfully added (Figure 11-4 on page 432).

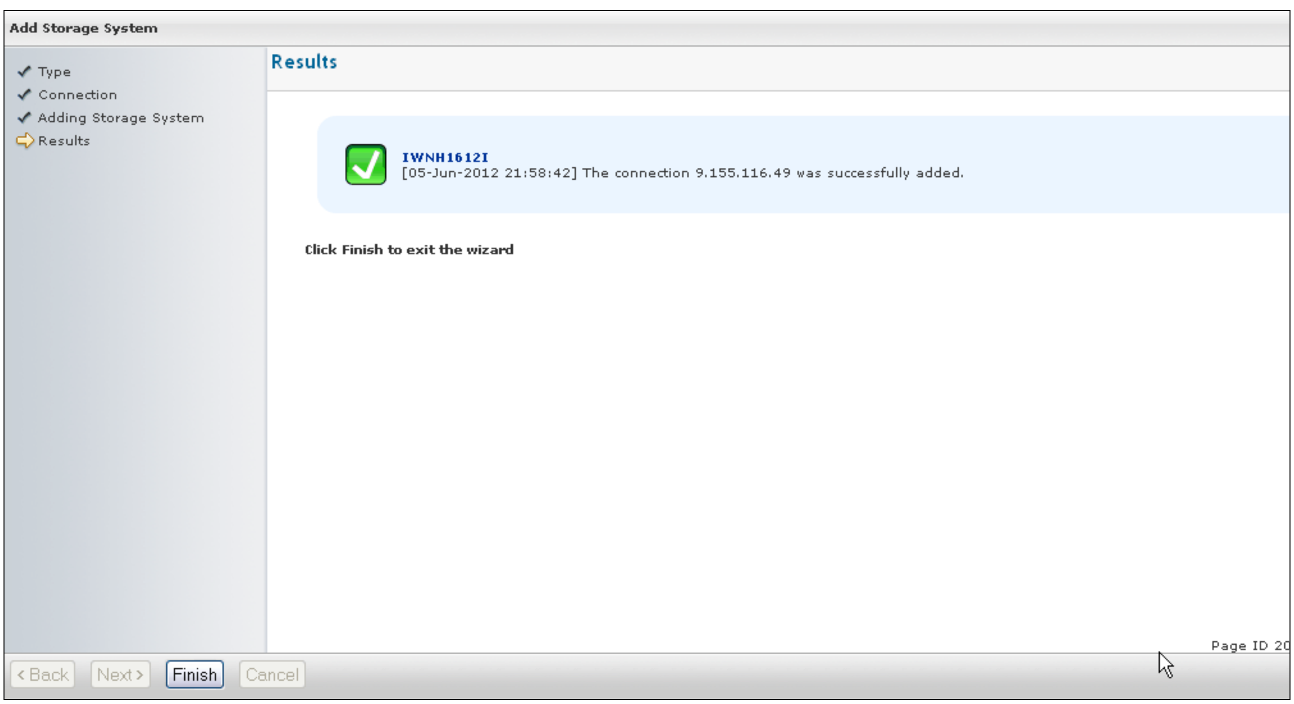


Figure 11-4 System was successfully added

Figure 11-5 shows the storage system is in the managed system now.

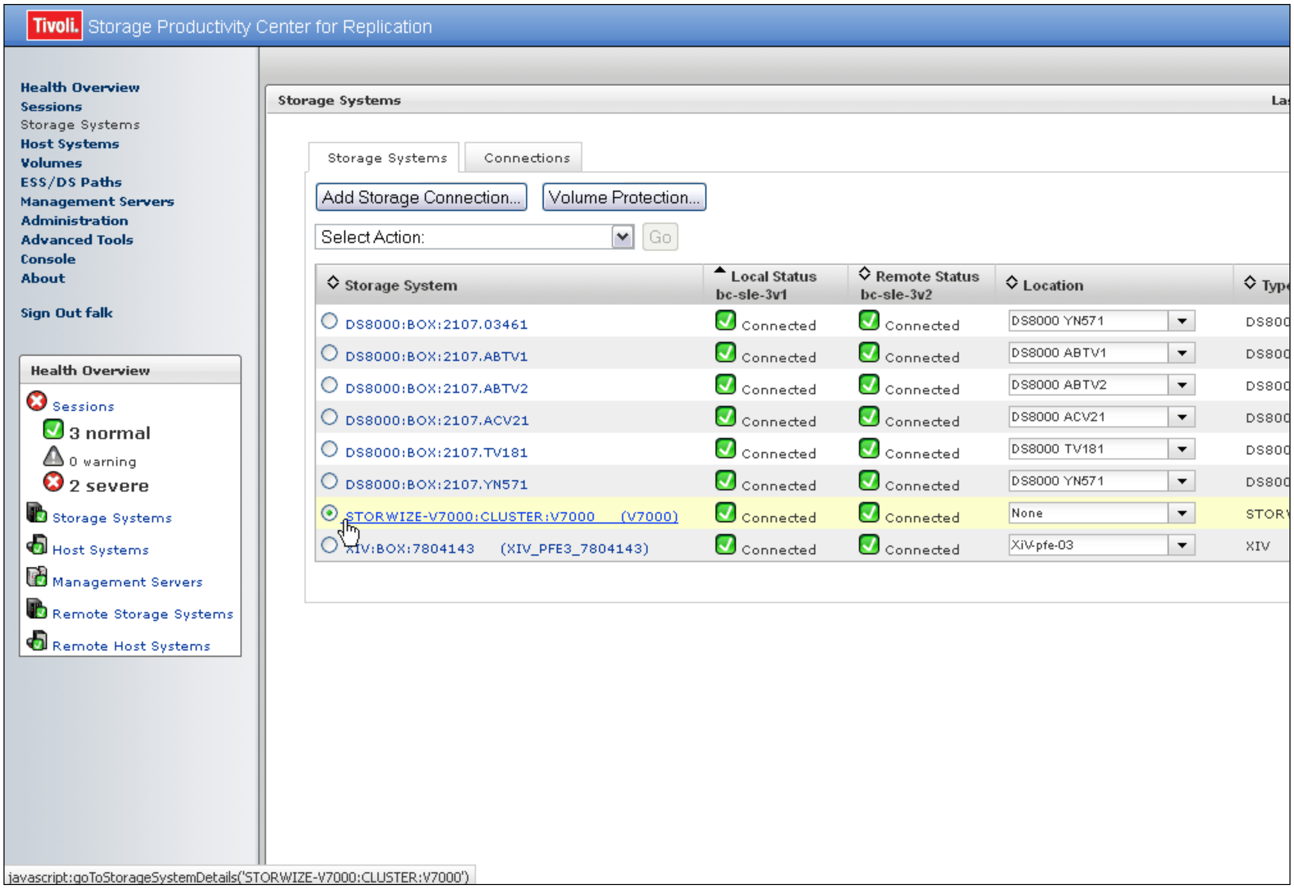


Figure 11-5 Storage Systems list

11.1.3 Configuring Copy Services relationships

Setting up Copy Services relationships in TPC-R is similar to using the SVC console to set up Copy Services relationships. The greatest difference is that all TPC-R relationships must be in a Consistency Group (called a "session" in TPC), even if you only want to create a single relationship.

To start the process, follow these steps:

1. Go to **Sessions** → **Create Session**. Select the type of the relationship, as shown in Figure 11-6.

   In addition, there are several session types with which you might not be familiar. We discuss Failover/Failback sessions in 11.1.4, "Failover/Failback Metro and Global Mirror sessions" on page 442 and the "with practice" sessions in 11.1.5, "Metro and Global Mirror with practice sessions" on page 443.

   For this example, a FlashCopy session is created.

2. On the next window, fill in a name for the session (this name will appear in the SVC as a Consistency Group) and fill in the background copy rate (0 is equivalent to NOCOPY).

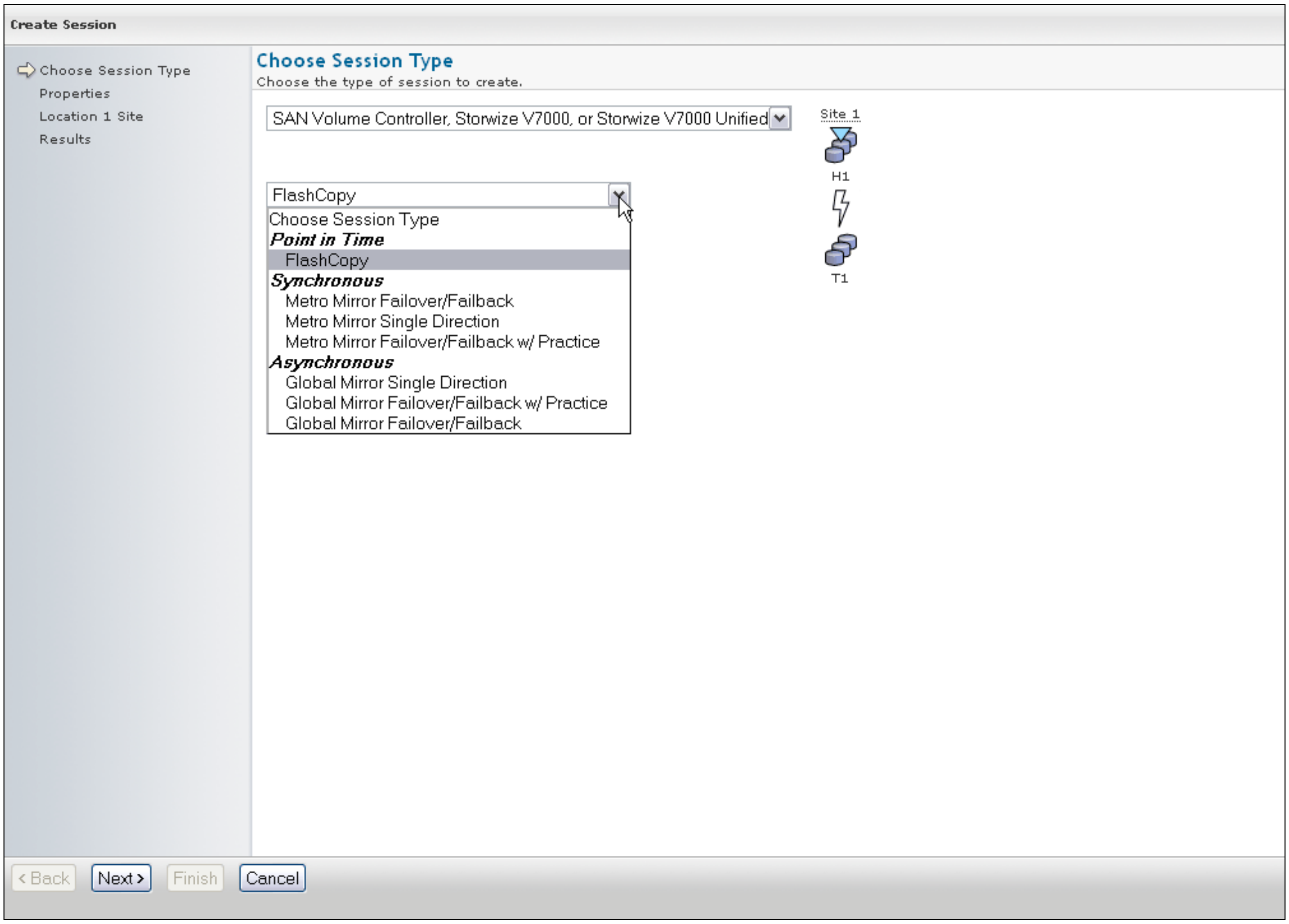


Figure 11-6 Session type selection

The sample session is shown in Figure 11-7 on page 435.

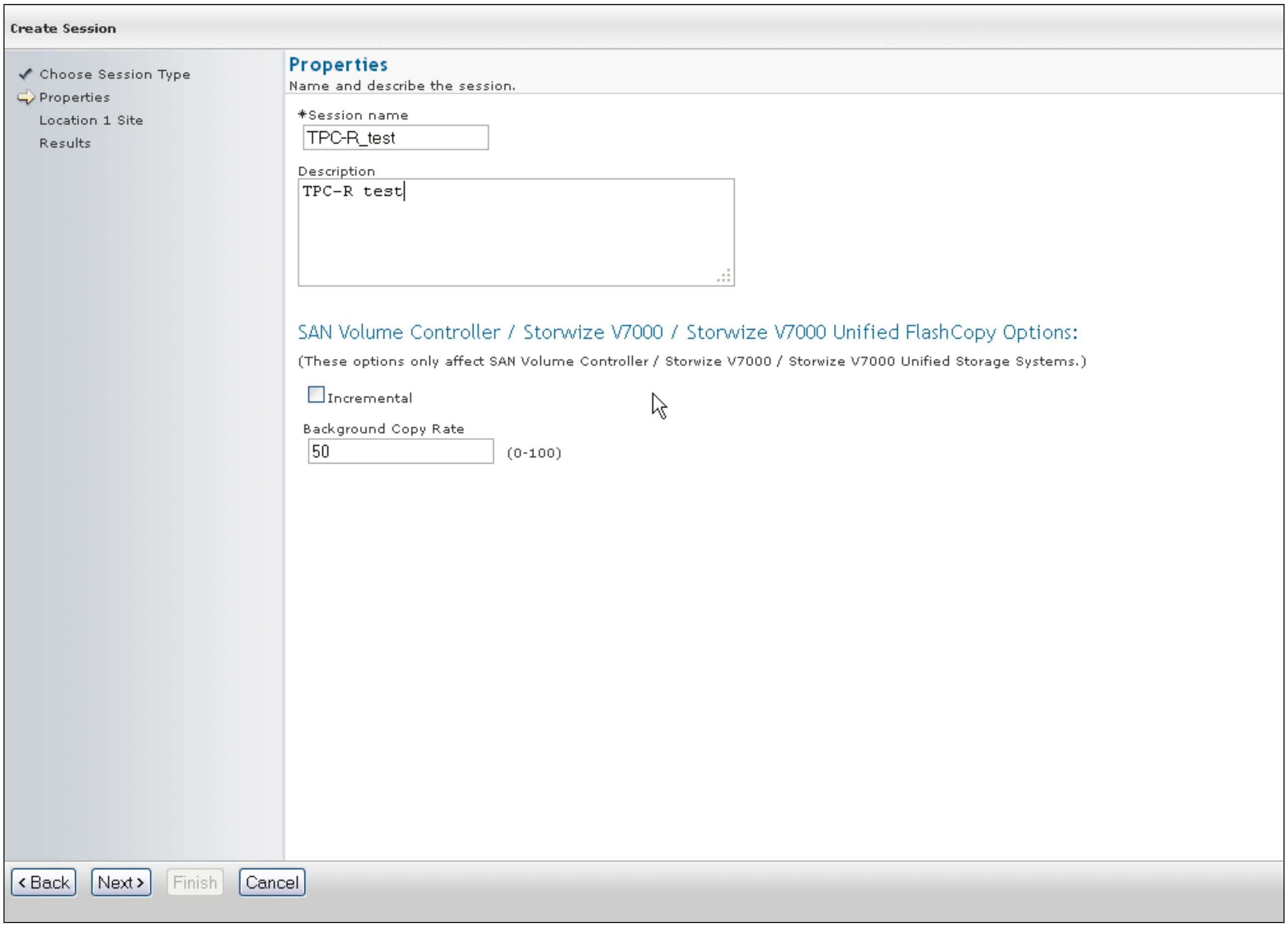


Figure 11-7 FlashCopy parameters in TPC-R

Select the location of the system and that finishes the process. Figure 11-8 on page 436 shows the completion window.

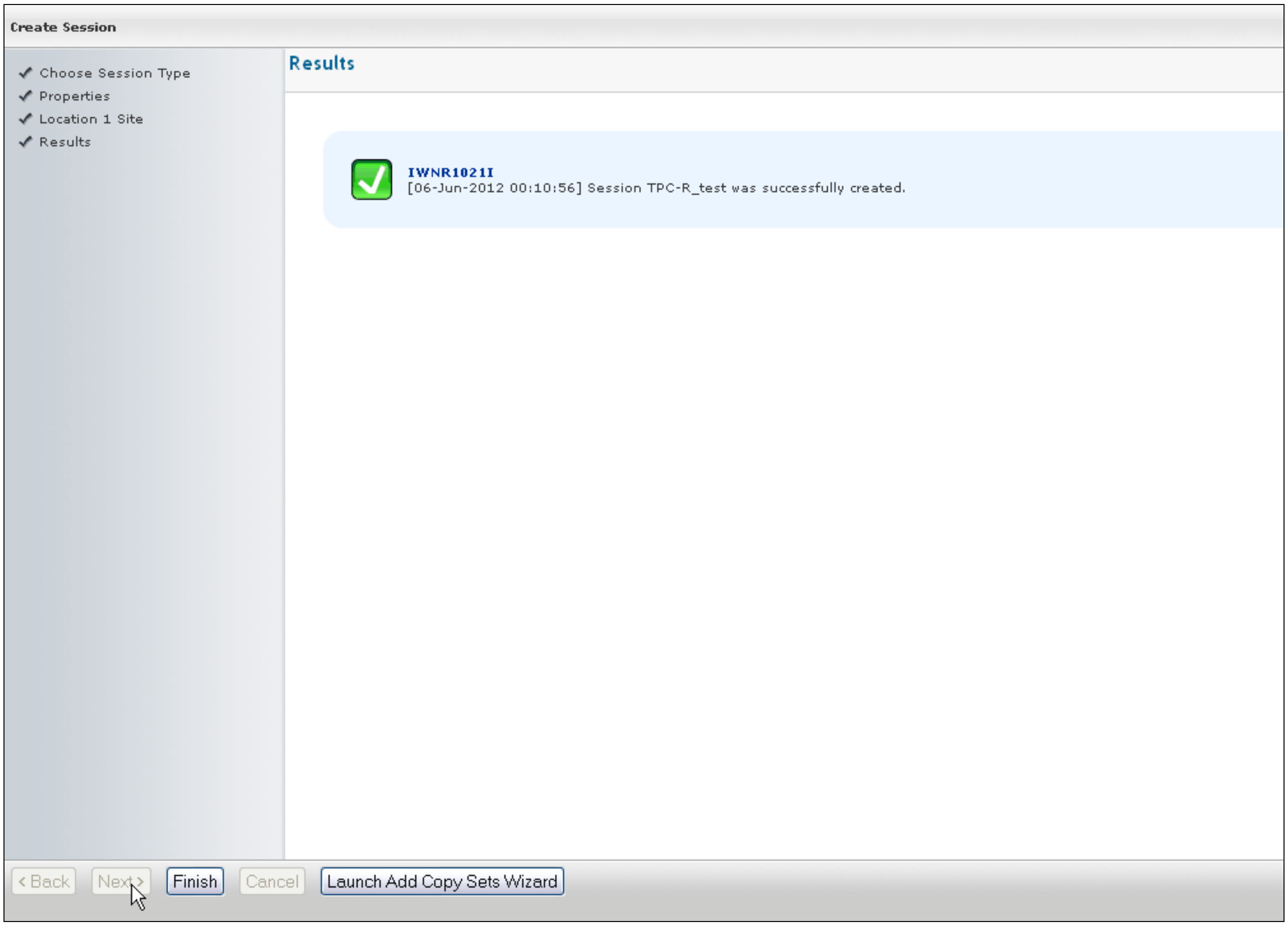


Figure 11-8 FlashCopy session completion

3. For the next step you typically add mappings called “copy sets” in TPC to your new session. You can click **Launch Add Copy Sets Wizard** on Figure 11-8.

   Alternatively, you can add copy sets from the Sessions window by selecting the session and then selecting **Add Copy Sets** from the drop-down menu, as shown in Figure 11-9 on page 437.

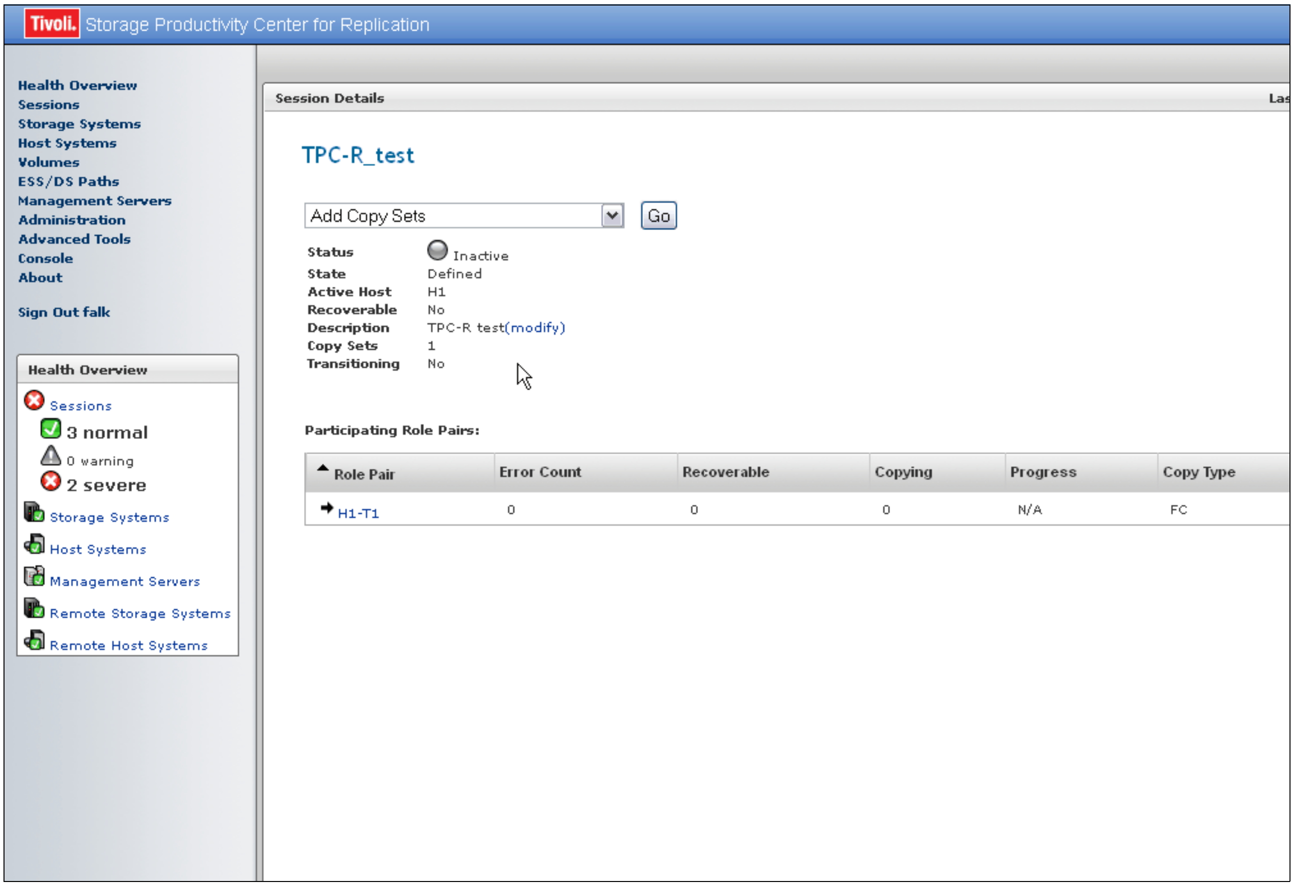


Figure 11-9 Create copy set from the Session window

4. On the initial creation window, you select the SVC cluster, I/O Group, and Volume. The parameters for the source volume of our sample copy set are shown in Figure 11-10 on page 438.

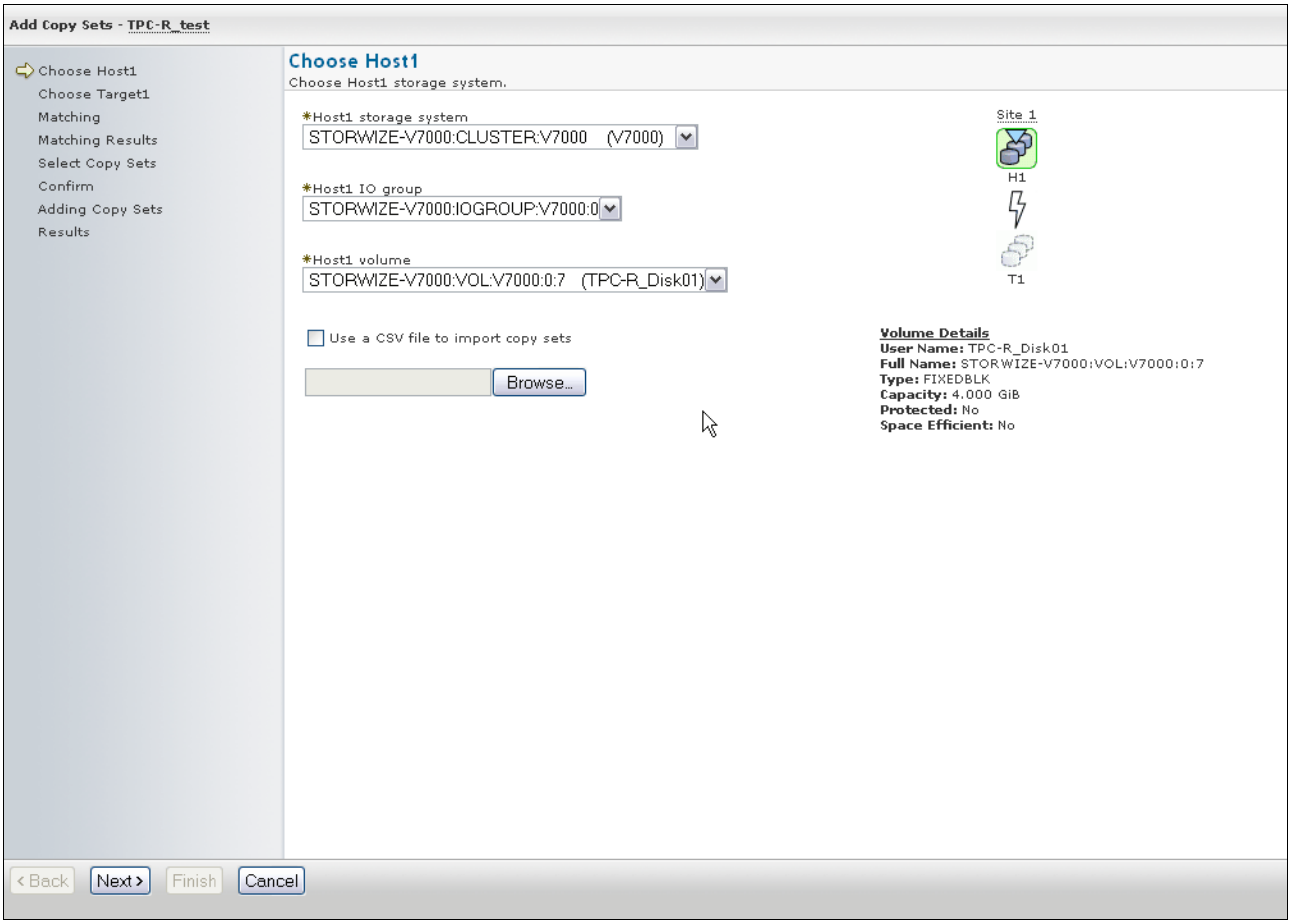


Figure 11-10 Source volume parameters

5. Repeat the process for the target volume. When you open the drop-down menu for target volumes the list is displayed with compatible volumes only, with the same size as the source volume. Click **Next**. The window that results is shown in Figure 11-11 on page 439.

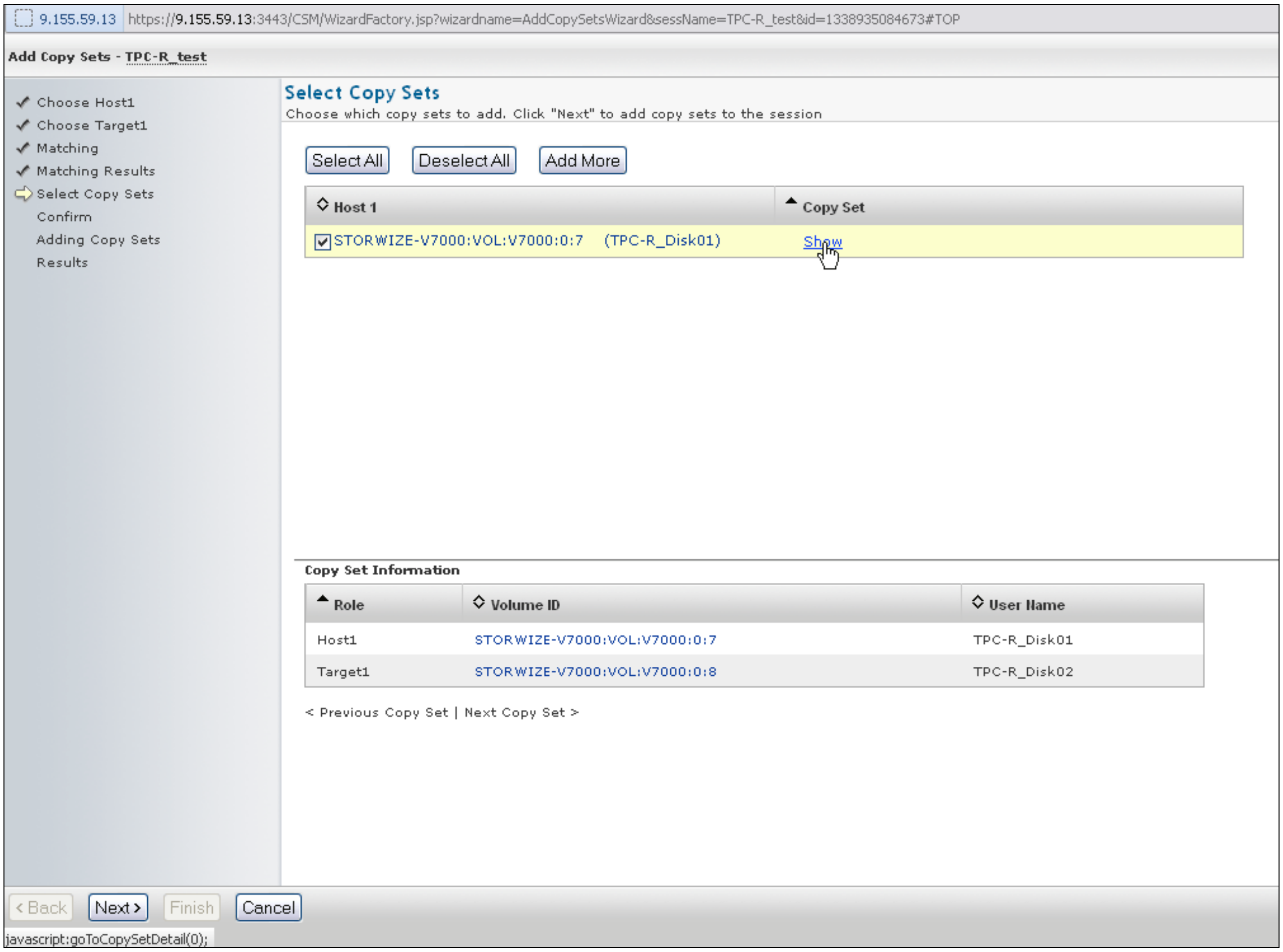


Figure 11-11 Add copy set result

6. From this point, you can add additional copy sets by clicking **Add More** at the top of the window. If you click **Show** under **Copy Set** you will obtain Copy Set details. You can end this process by clicking **Next** to complete the creation of the session.

7. In our case, we added a total of two copy sets. At the end, we are returned to the session window with our successfully created FlashCopy session, as shown in Figure 11-12 on page 440.

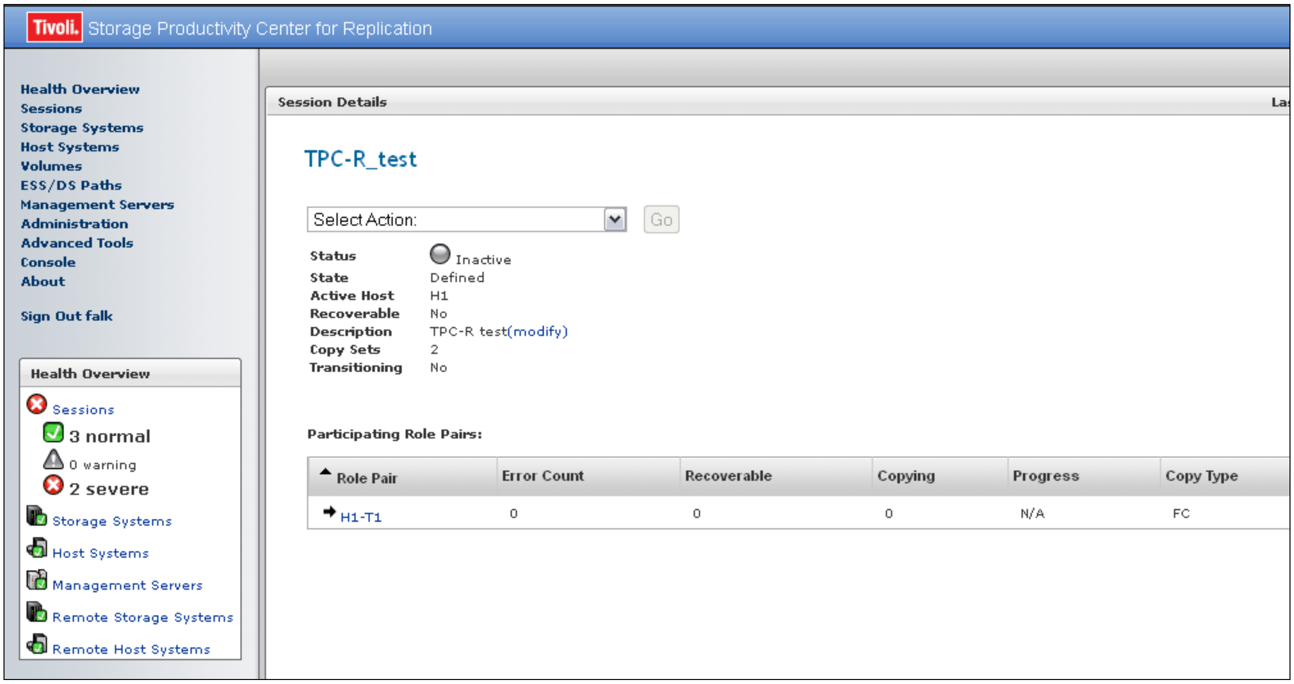


Figure 11-12 FlashCopy session created

It is interesting to note that if we go to the SVC console at this point we do *not* see the mappings listed, because TPC-R does not actually create the mappings (or Consistency Group) until they are used for the first time.

- There are several options listed in the drop-down menu to use the FlashCopy session: Flash and Start. The option names are explained here:
  - "Start" in TPC-R means "Prepare" in the SVC.
  - "Flash" in TPC-R means "Start" in the SVC.
  - To stop a FlashCopy session use "Terminate" in TPC-R, which is equivalent to "Stop" in the SVC.

After we select **Flash**, we are presented with the warning dialog that is shown in Figure 11-13 on page 441. After selecting **Yes**, the FlashCopy occurs.

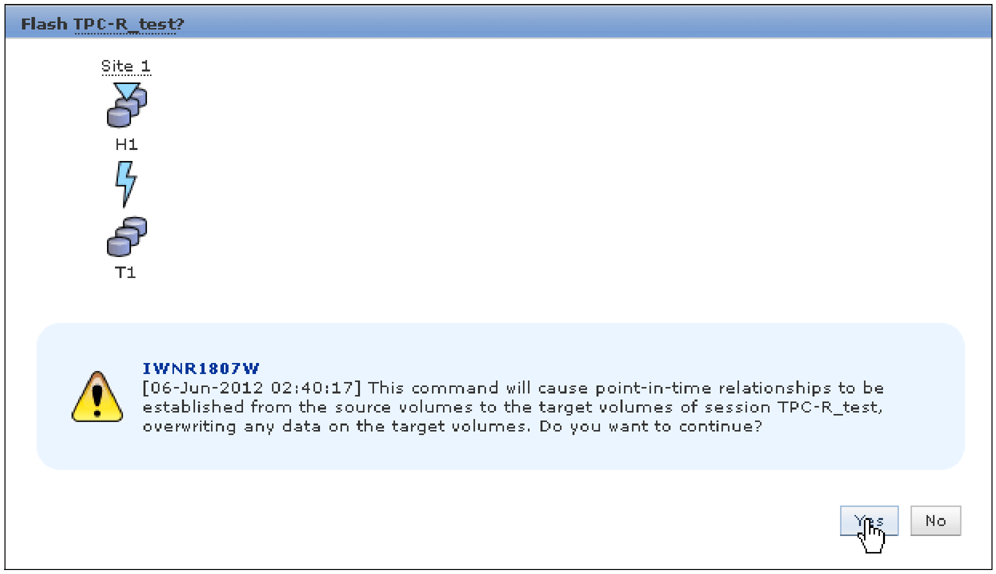


Figure 11-13 FlashCopy warning dialog

Figure 11-14 shows the final state of the session.

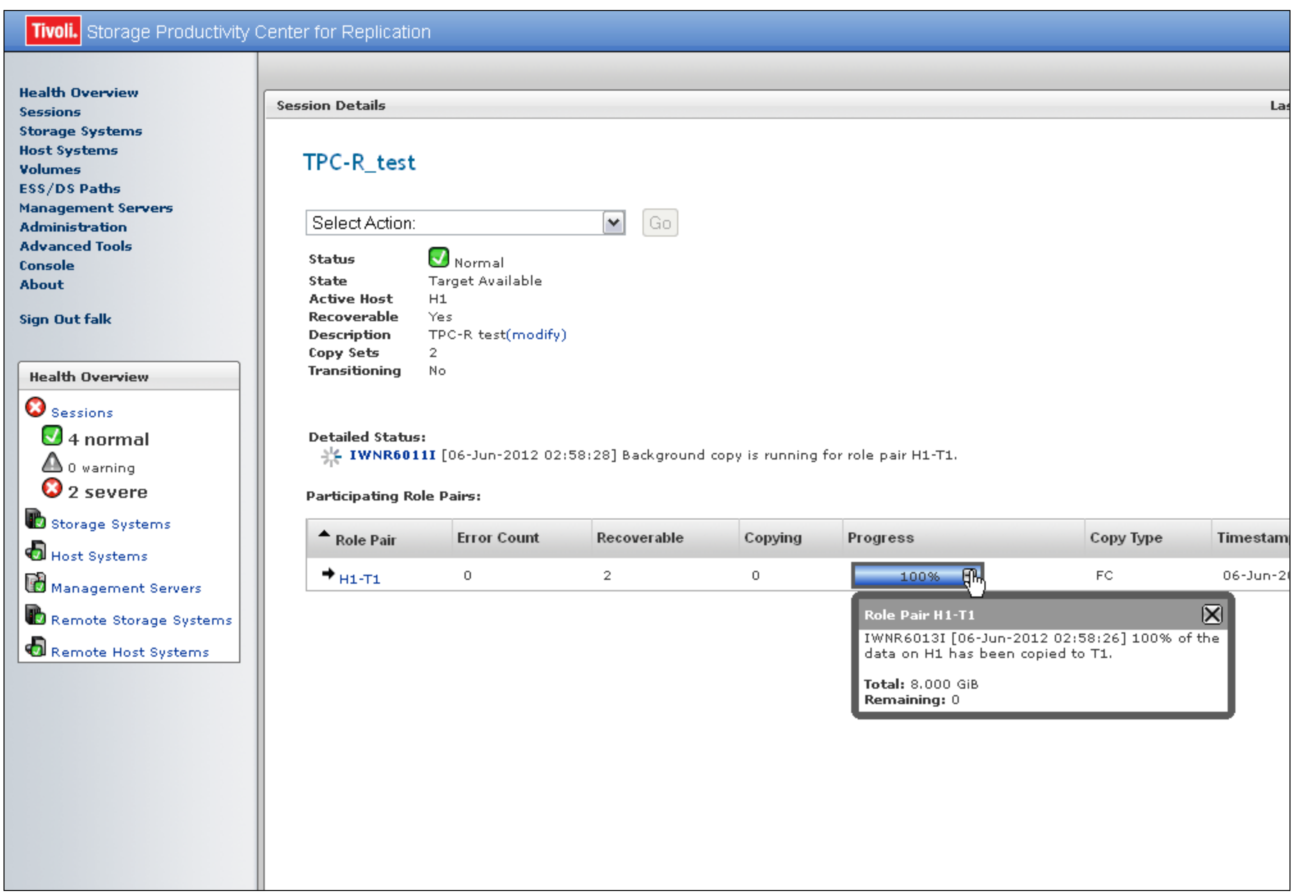


Figure 11-14 FlashCopy success

Now, if we go to the SVC GUI, we can see the mappings that we created (and in the *Copying* state at the time of the capture). Figure 11-15 on page 442 shows the mappings that we created. Note that the names of the relationships are automatically generated, and they do not use the names that we assigned to them in the TPC-R GUI.

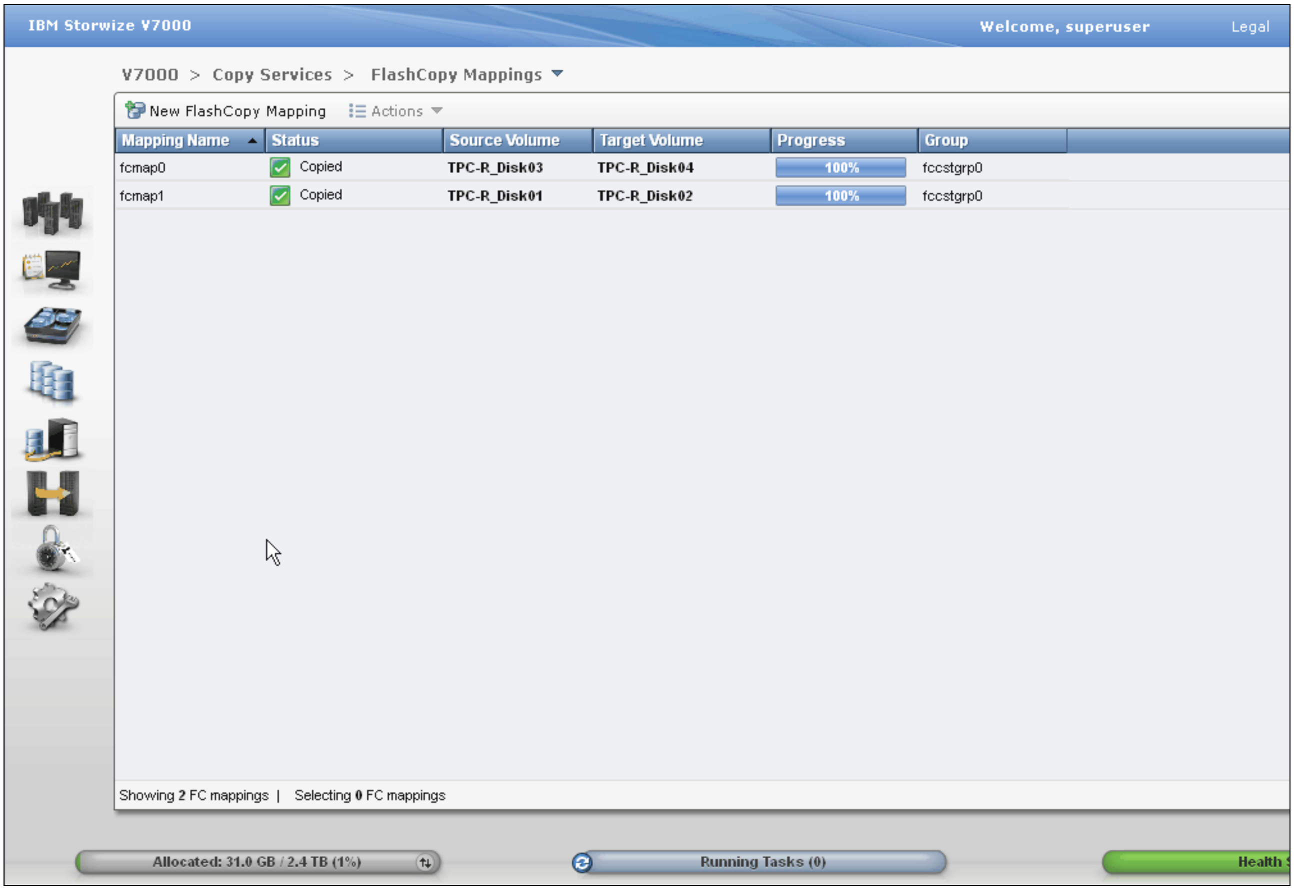


Figure 11-15 SVC GUI with TPC-R FlashCopy mappings

11.1.4 Failover/Failback Metro and Global Mirror sessions

The Failover/Failback feature is available in TPC-R. This feature allows you to use TPC-R to enable the destination disks for writing, and later reverse the copy direction. However, you can perform these same actions by using the SVC console.

To use TPC-R to perform this failover and failback, you carry out several operations in sequence:

1. **Suspend** the session (assuming a loss of connectivity has not suspended it for you).
2. Select **Recover** to make the destination site volumes writable.
3. After the primary site has been restored, select **Enable Copy to Site 1** to reverse the copy direction.
4. Use **Start H2** → **H1** to copy the data back to the Primary site. When it is finished successfully use **Suspend** the session.
5. Select **Recover** again to enable the Primary volumes for writing again.
6. Select **Enable Copy to Site 2** to reverse the copy direction again.
7. Select **Start H1** → **H2** to restart the copy operation.

11.1.5 Metro and Global Mirror with practice sessions

TPC-R defines a new type of Copy Services relationship: the “with practice” relationship type. This type enables you to perform disaster recovery tests on your remote disks. This feature generates a FlashCopy relationship to several *practice* volumes at your destination site so that you can practice your data recovery with the servers located at the destination.

You can just as easily create a separate FlashCopy session that included the same pair of volumes, although this will display a warning dialog informing you that one of the disks is already in a Copy Services relationship.

To create a with practice session, you need to designate three volumes that are called `H1`, `I2`, and `H2` by TPC-R. These volumes are slightly confusing, because in with practice sessions, your mirroring destination is I2, not H2. H2 is the practice volume that serves as the FlashCopy destination.

As with the other relationship types, no relationships will be created on the SVC until they are used for the first time. In the case of practice sessions, the FlashCopy relationships will not be created until you Flash for the first time; they are not created when the mirroring relationship is started.

11.1.6 Using CSV files during Copy Set creation

Unlike with the SVC GUI, there are no filters for the volume name, so using the TPC-R GUI for a large configuration can be tedious. Fortunately, you can add Copy Sets to a Session by importing a comma-separated value (CSV) file. The CSV file must take the format shown in Example 11-1.

Example 11-1 TPC FlashCopy CSV file

```
H1,T1
SVC:VOL:9.43.86.119:0:0,SVC:VOL:9.43.86.119:0:3
SVC:VOL:9.43.86.119:0:1,SVC:VOL:9.43.86.119:0:4
SVC:VOL:9.43.86.119:0:2,SVC:VOL:9.43.86.119:0:5
```

Note the following points:

- As shown in Example 11-1, the first line of an ordinary FlashCopy CSV contains H1,T1.
- For a Metro Mirror or Global Mirror session, use H1,H2.
- For a practice session, use H1,I1,H2.

For non-practice relationships, the remaining lines each take the following format:

```
SVC:VOL:<SVC_IP_ADDRESS>:<SOURCE_SVC_IO_GROUP>:<SOURCE_VOLUME_NUMBER>,SVC:VOL:<SVC
_IP_ADDRESS>:<SVC_TARGET_IO_GROUP>:<TARGET_VOLUME_NUMBER>
```

For practice relationships, use:

```
SVC:VOL:<SVC_IP_ADDRESS>:<SOURCE_SVC_IO_GROUP>:<SOURCE_VOLUME_NUMBER>,SVC:VOL:<SVC
_IP_ADDRESS>:<SVC_INTERMEDIATE_IO_GROUP>:<INTERMEDIATE_VOLUME_NUMBER>SVC:VOL:<SVC_
IP_ADDRESS>:<SVC_TARGET_IO_GROUP>:<TARGET_VOLUME_NUMBER>
```

You might be wondering where the volume numbers come from. They can be obtained by retrieving the output of **`lsvol`** from the TPC-R command line. Example 11-2 on page 444 shows example output. All of our command-line examples assume that you have configured TPC-R to automatically log in.

The details for configuring TPC-R to automatically log in are in the TPC Information Center:

http://publib.boulder.ibm.com/infocenter/tivihelp/v4r1/index.jsp

Example 11-2 lsvol output

```
C:\Program Files\IBM\replication>csmcli lsvol
TotalStorage Productivity Center for Replication Command Line Interface (CLI)
 Copyright 2007 IBM Corporation
 Version: 3.4
 Build: c121-080327b

Name            ID                       Dev                   Dev Type Protected
Space Efficient
==================================================================================
=============
vd-prod-12-0002 SVC:VOL:9.43.86.131:0:3 2145-9.43.86.131-IBM SVC      No         No
vd-prod-12-0001 SVC:VOL:9.43.86.131:0:2 2145-9.43.86.131-IBM SVC      No         No
target-fa       SVC:VOL:9.43.86.131:0:1 2145-9.43.86.131-IBM SVC      No         No
source-se       SVC:VOL:9.43.86.131:0:0 2145-9.43.86.131-IBM SVC      No         No
vd-prod-11-0002 SVC:VOL:9.43.86.119:0:9 2145-9.43.86.119-IBM SVC      No         No
vd-prod-11-0001 SVC:VOL:9.43.86.119:0:8 2145-9.43.86.119-IBM SVC      No         No
ITSO_MM_M000002 SVC:VOL:9.43.86.119:0:7 2145-9.43.86.119-IBM SVC      No         No
ITSO_MM_M000001 SVC:VOL:9.43.86.119:0:6 2145-9.43.86.119-IBM SVC      No         No
SenegalFC0003   SVC:VOL:9.43.86.119:0:5 2145-9.43.86.119-IBM SVC      No         No
SenegalFC0002   SVC:VOL:9.43.86.119:0:4 2145-9.43.86.119-IBM SVC      No         No
SenegalFC0001   SVC:VOL:9.43.86.119:0:3 2145-9.43.86.119-IBM SVC      No         No
Senegal0003     SVC:VOL:9.43.86.119:0:2 2145-9.43.86.119-IBM SVC      No         No
Senegal0002     SVC:VOL:9.43.86.119:0:1 2145-9.43.86.119-IBM SVC      No         No
Senegal0001     SVC:VOL:9.43.86.119:0:0 2145-9.43.86.119-IBM SVC      No         No
vdisk9          SVC:VOL:9.43.86.117:0:9 2145-9.43.86.117-IBM SVC      No         No
vdisk8          SVC:VOL:9.43.86.117:0:8 2145-9.43.86.117-IBM SVC      No         No
siam1           SVC:VOL:9.43.86.117:0:7 2145-9.43.86.117-IBM SVC      No         No
vdisk6          SVC:VOL:9.43.86.117:0:6 2145-9.43.86.117-IBM SVC      No         No
vdisk5          SVC:VOL:9.43.86.117:0:5 2145-9.43.86.117-IBM SVC      No         No
siam0           SVC:VOL:9.43.86.117:0:4 2145-9.43.86.117-IBM SVC      No         No
vdisk3          SVC:VOL:9.43.86.117:0:3 2145-9.43.86.117-IBM SVC      No         No
vdisk2          SVC:VOL:9.43.86.117:0:2 2145-9.43.86.117-IBM SVC      No         No
Dio_imagemode   SVC:VOL:9.43.86.117:0:0 2145-9.43.86.117-IBM SVC      No         No
```

You can also find the volume number by looking at the Volume ID column (not the UID column) in the output of the **`svcinfo lsvdisk`** command in the SVC command-line interface (CLI).

11.1.7 TPC for Replication CLI

As with the SVC, TPC-R has a full-featured CLI. You can perform all routine operations with the CLI that you can perform with the GUI. You can obtain full details of the syntax for the CLI in the TPC documentation, which is located at:

http://publib.boulder.ibm.com/infocenter/tivihelp/v4r1/index.jsp

11.1.8 TPC-R Live demo

For TPC-R Live demonstration, refer to IBM European Storage Competence Center at Mainz, Germany:

http://ibm-vbc.centers.ihost.com/briefingcenter/mainz

For a full demonstration list, refer to the following site:

https://www-927.ibm.com/servers/eserver/storageplaza/BERT.nsf/solutions

For remote demo request refer to the following site:

https://www-927.ibm.com/servers/eserver/storageplaza/BERT.nsf/pages/requestMAZ?OpenDocument

For comprehensive, in-depth technology briefings, product demonstrations and solution workshops for IBM clients and IBM Business Partners wanting product expertise on IBM Systems and Storage solutions, refer to IBM Executive Briefing Center:

http://www-03.ibm.com/systems/services/briefingcenter/

11.1.9 IBM Tivoli Storage Productivity Center V5.1

IBM Tivoli Storage Productivity Center V5.1, announced in June 2012, offers storage infrastructure management that helps optimize storage management by centralizing, simplifying, automating, and optimizing storage tasks associated with storage systems, data disaster recovery, storage networks, and capacity management.

IBM Tivoli Storage Productivity Center V5.1 products include:

- IBM Tivoli Storage Productivity Center V5.1
- IBM Tivoli Storage Productivity Center *Select* Edition V5.1

Tivoli Storage Productivity Center V5.1 is designed to provide device management capabilities such as automated system discovery, provisioning, data replication, configuration, and performance monitoring for storage systems and storage networks.

Tivoli Storage Productivity Center *Select* Edition V5.1 offers the same features as Tivoli Storage Productivity Center V5.1 but at attractive entry-level pricing for operations with smaller capacities. It is licensed per storage device, such as disk controllers and their respective expansion units.

The following items are new in Tivoli Storage Productivity Center V5.1:

- A next-generation, web-based user interface that is designed to offer ease of use. The new user interface provides a common look and feel that is based on the current user interfaces for IBM XIV Storage System, IBM Storwize V7000, and IBM System Storage SAN Volume Controller. It enables quick access to key storage assets, status, and performance information.
- IBM Cognos-based reporting that helps create and integrate custom reports on capacity, performance, and utilization.
- Replication features that include support for external failover by external products, allowing users to define Warning and Severe thresholds for recovery point objective (RPO) alerts, and providing the ability to export replication history to a table for analysis.

- An InstallAnywhere wizard to help simplify installation and post-installation configuration to help accelerate administrator time-to-value.

Statement of direction

IBM intends to release a new storage software offering that combines storage virtualization, storage resource management, and application aware snapshots. IBM also intends to provide enhancements to these capabilities that can generate recommendations, and enable automated movement and placement of data across storage systems based on policy and intelligent algorithms to better optimize for price, performance, and access needs across the storage environment.

IBM intends to adapt its advanced administration GUI to IBM Tivoli Storage Manager, simplifying administration. This intuitive GUI approach is already being used across the IBM storage portfolio of software and systems: Tivoli Storage Productivity Center, IBM System Storage SAN Volume Controller, IBM XIV Storage System, IBM Storwize V7000 Unified, and IBM Scale Out Network Attached Storage (SONAS). Customers can leverage this user interface consistency to simplify the management of various systems within their data center such as unified recovery, enterprise storage administration, and individual storage systems.

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion. Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision. The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

11.2 IBM Tivoli Storage FlashCopy Manager

Today’s applications and servers operate 24 hours a day, 7 days a week. Enterprises cannot afford backup downtime for applications or data. Backups need to be “hot” and without application disruptions.

IBM SVC Replication Family Services provide “crash-consistent” copies of data. Data is consistent at a storage level. Many business-critical applications such as relational databases keep some of the data in memory. This data should be flushed from memory and committed to the disks prior to making a “crash-consistent” copy. It is also an “application-consistent” copy and guarantees that the data copy is perfect for disaster recovery, cloning or other post-volume copy task.

IBM Tivoli Storage FlashCopy Manager provides a high level of protection and helps to obtain “application-consistent” copies for a quick and reliable disaster recovery/cloning process for IBM DB2, Oracle, SAP, Microsoft SQL Server and Exchange applications. IBM Tivoli Storage FlashCopy Manager integrates with IBM System Storage SAN Volume Controller and IBM Storwize V7000, and provides quick and easy way to crate and manage “application-consistent” copies.

For a true disaster recovery test, it is advisable to use “application-consistent” copies and avoid scenarios that are not close to real situations. For example, do not shut down the production servers or applications prior to the test. Be aware that IBM Tivoli Storage Manager for Advanced Copy Services has been replaced and integrated into IBM Tivoli Storage FlashCopy Manager.

An overview of this topic is available at:

http://www.youtube.com/watch?v=2MPUrOOo2Sw

For further details, refer to IBM Tivoli Storage FlashCopy Manager at:

http://www-01.ibm.com/software/tivoli/products/storage-flashcopy-mgr/

11.2.1 IBM Tivoli Storage FlashCopy Manager for Windows

Tivoli Storage FlashCopy Manager provides the tools and information needed to create and manage volume-level snapshots of Microsoft SQL Server and Microsoft Exchange Server data. These snapshots are created while these applications (with volume data) remain online. Tivoli Storage FlashCopy Manager provides support to create and manage volume-level snapshots for File Systems and Custom Applications. It uses Microsoft Volume Shadow Copy Services (VSS) and IBM storage hardware snapshot technology to protect your business-critical data.

11.2.2 Microsoft Exchange Server backup and restore with IBM Tivoli Storage FlashCopy Manager for Windows

The following example shows how to back up and restore a Microsoft Exchange database and a particular mailbox with IBM Tivoli Storage FlashCopy Manager and IBM Storwize V7000 FlashCopy Services.

In our example we have an Exchange mail server and two mail boxes named Bob and Ann. We have exchanged several e-mails between them as shown in Figure 11-16.

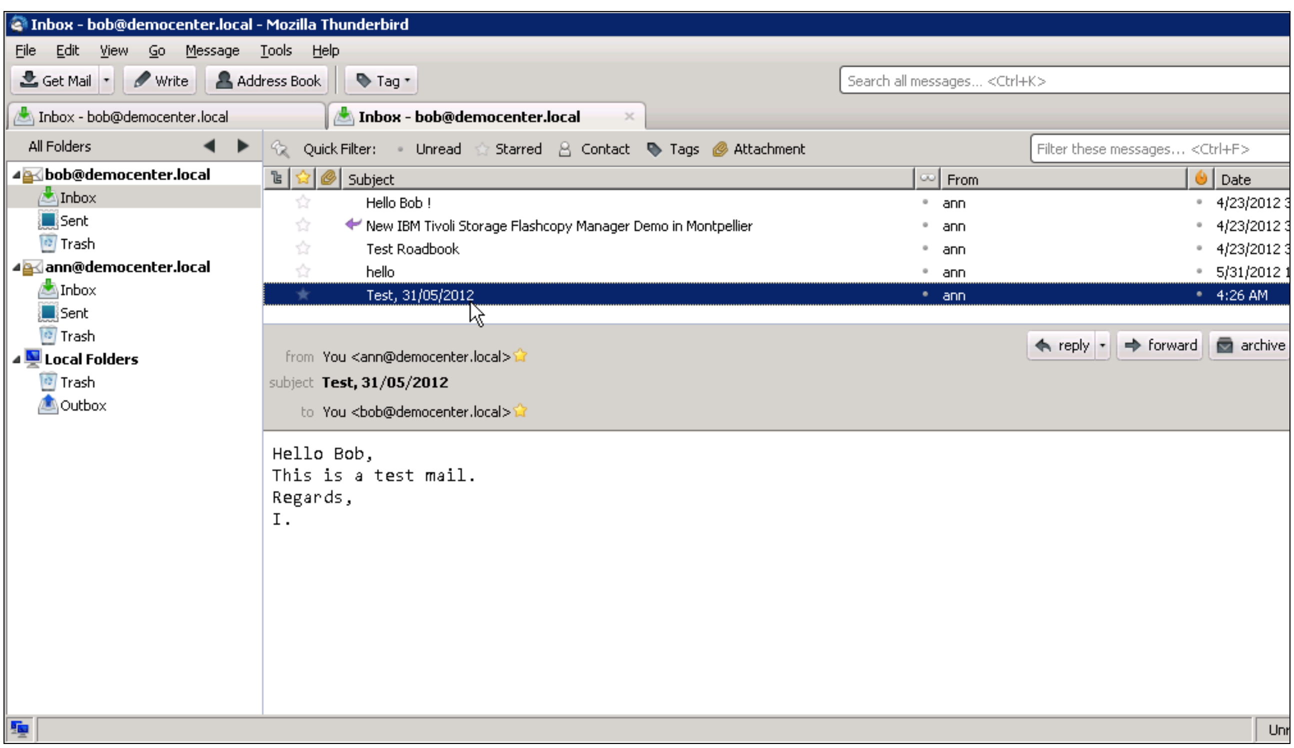


Figure 11-16 User mailboxes

We are ready to perform a backup so we open the Tivoli Storage FlashCopy Manager console, select **Exchange Database**, then right-click and select **Full Backup** as shown in Figure 11-17.

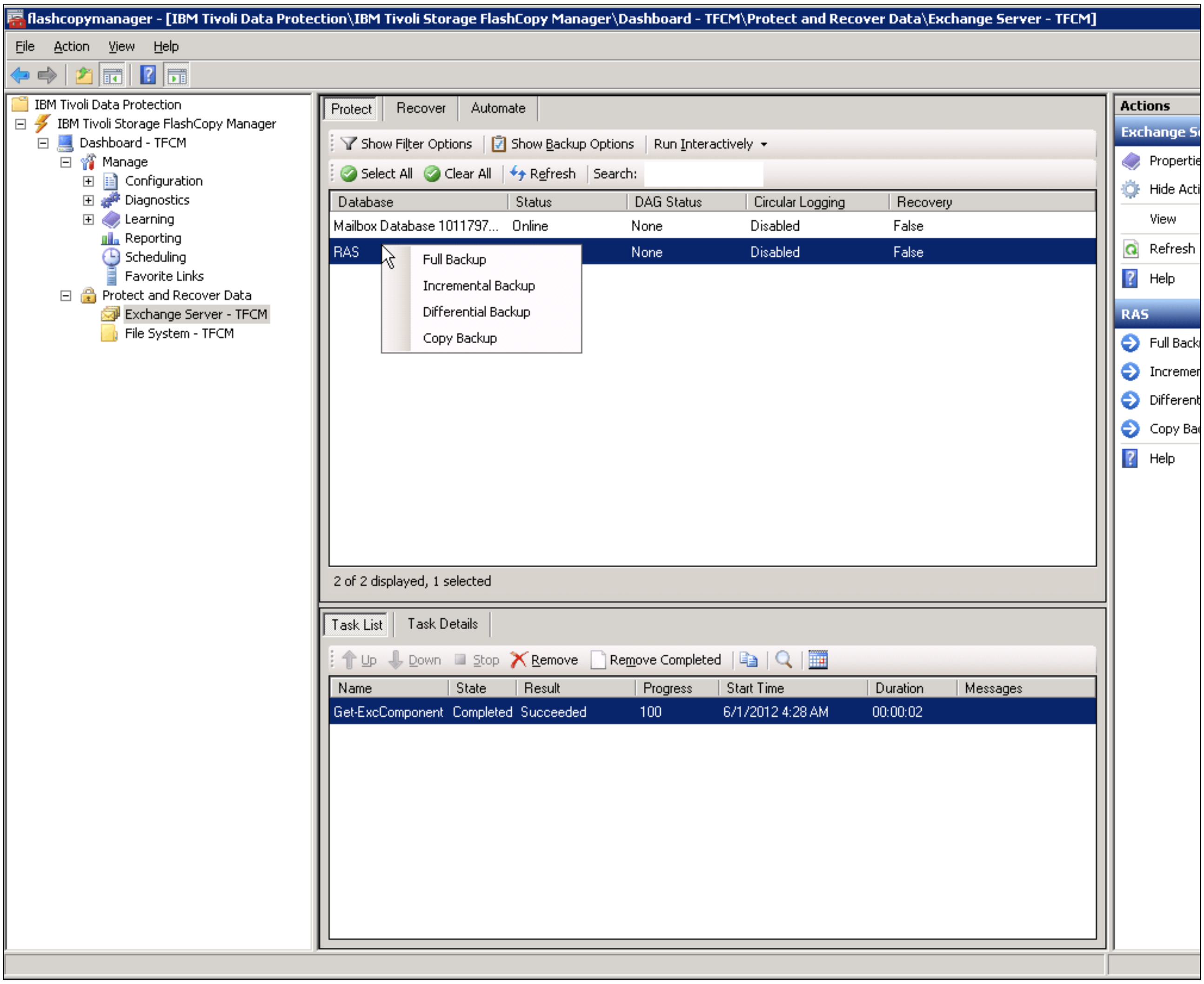


Figure 11-17 Exchange Database Full Backup

If you check IBM Storwize V7000 you will see that FlashCopy Services are in progress from **Source** to **Target** Volumes for Exchange Data and Log Volumes as shown in Figure 11-18 on page 449.

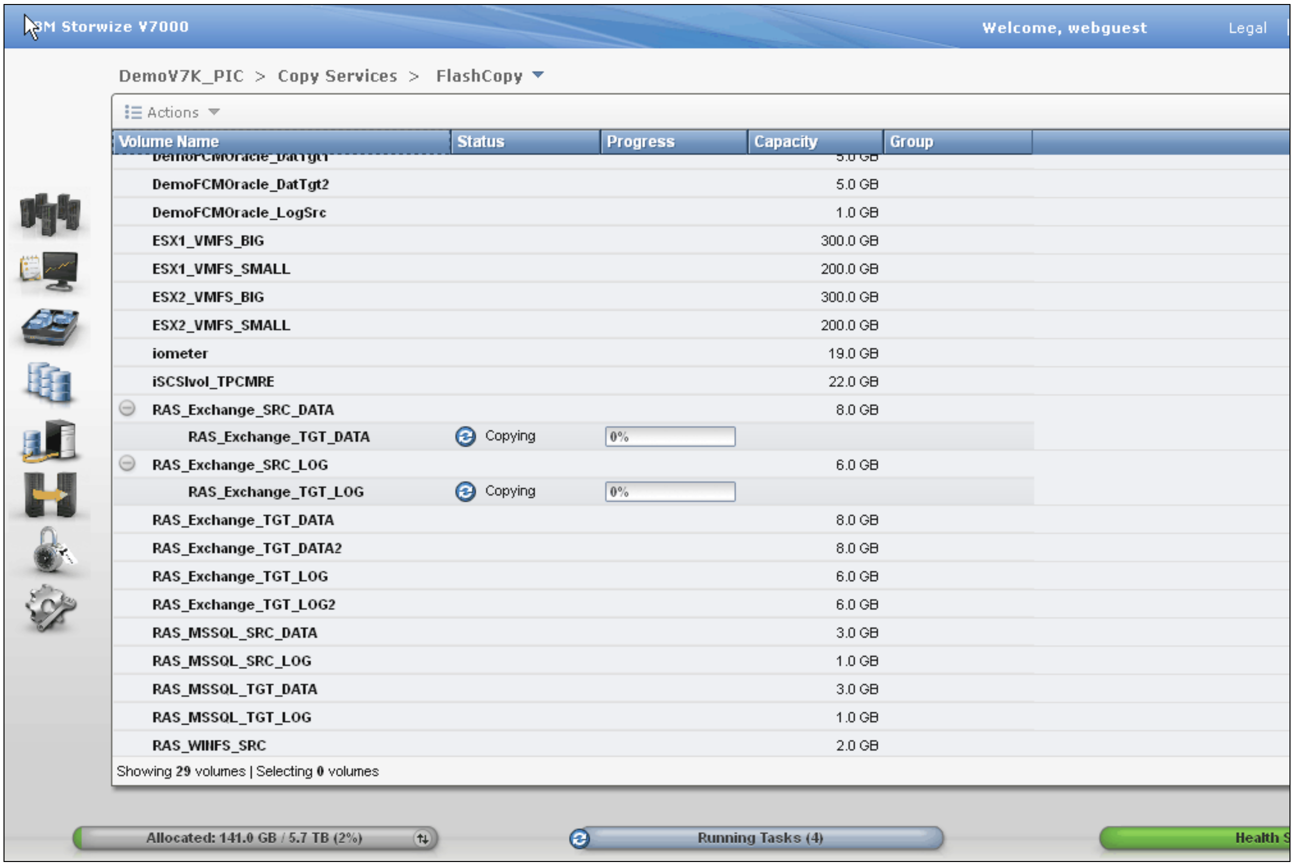


Figure 11-18 IBM Storwize V7000 FlashCopy Services are in progress

You can also use the **ibmvcfg.exe** utility to get the status from the Exchange Server as shown in Example 11-3.

Example 11-3 The ibmvcfg.exe utility and status information

```
c:\Program Files\IBM\Hardware Provider for VSS-VDS>ibmvcfg list infc
Physical host.
Provider Type is SVC Pegasus 6.3.
Listing flashcopy relationship(s)...

                                        <Please use the target vol. name to del fc map>
FC ID FC Name          Src ID  Src Name          Tgt ID  Tgt Name          InFC    Cp.Rate Cp%  Cl.Ra te Cl%
----------------------------------------------------------------------------------------------------------
0     fcmap0           13      RAS_Exchange_SRC_DATA 15      RAS_Exchange_TGT_DATA false   50      4    50      100
1     fcmap1           14      RAS_Exchange_SRC_LOG 16      RAS_Exchange_TGT_LOG false   50      6 50      100

Total: 2 pair(s) of flashcopy relationship.

c:\Program Files\IBM\Hardware Provider for VSS-VDS>
```

Tivoli Storage FlashCopy Manager Dashboard displays the Status Summary for the tasks performed (Figure 11-19 on page 450).

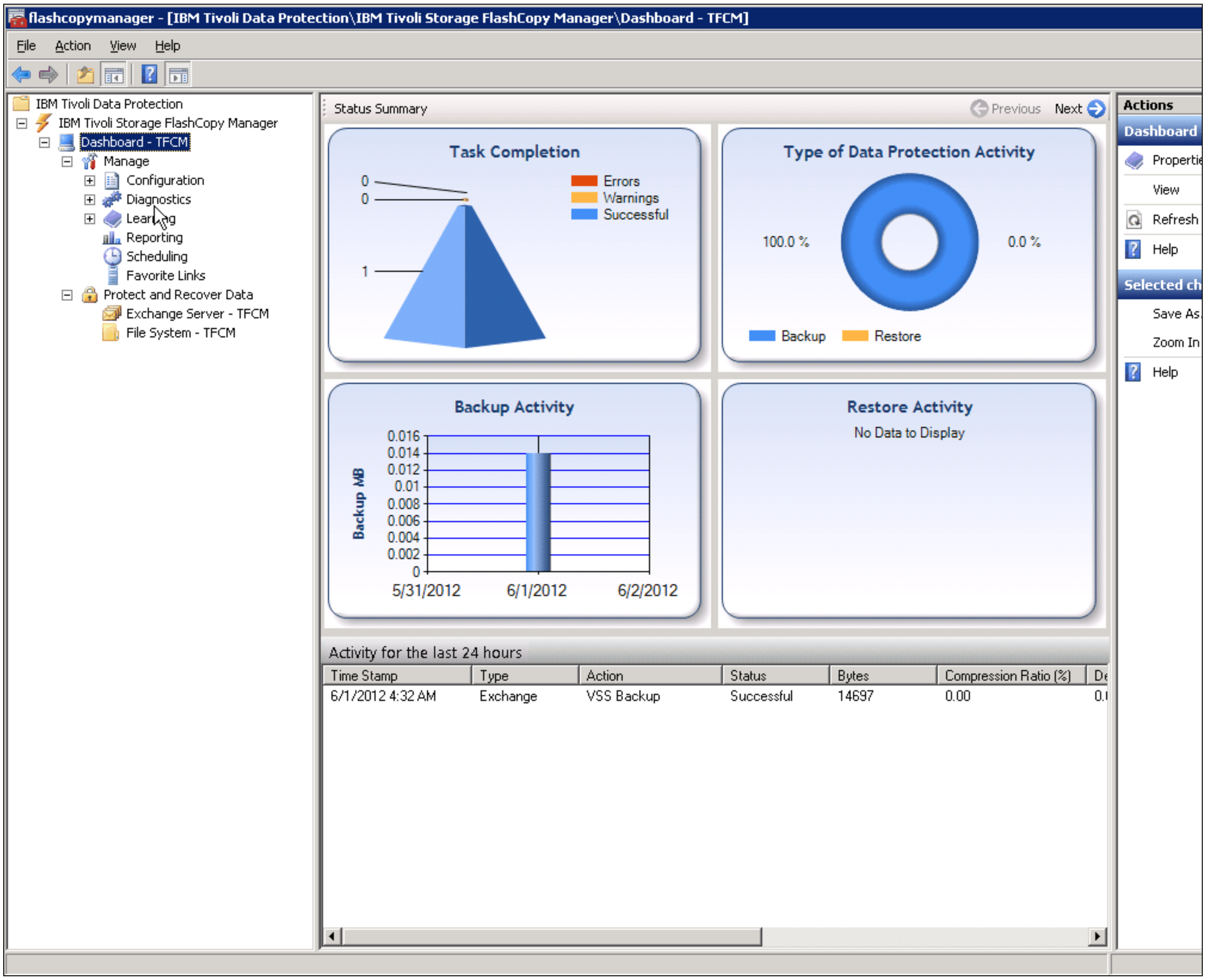


Figure 11-19 Tivoli Storage FlashCopy Manager Dashboard

In our case we delete all the messages in Bob's inbox, initiate a restore on a database level, and restore the entire Exchange Database (Figure 11-20 on page 451).

Check that the **Mount Database After Restore** option is **True**. This will fully automate the restore process and you will get all data restored and automatically mounted.

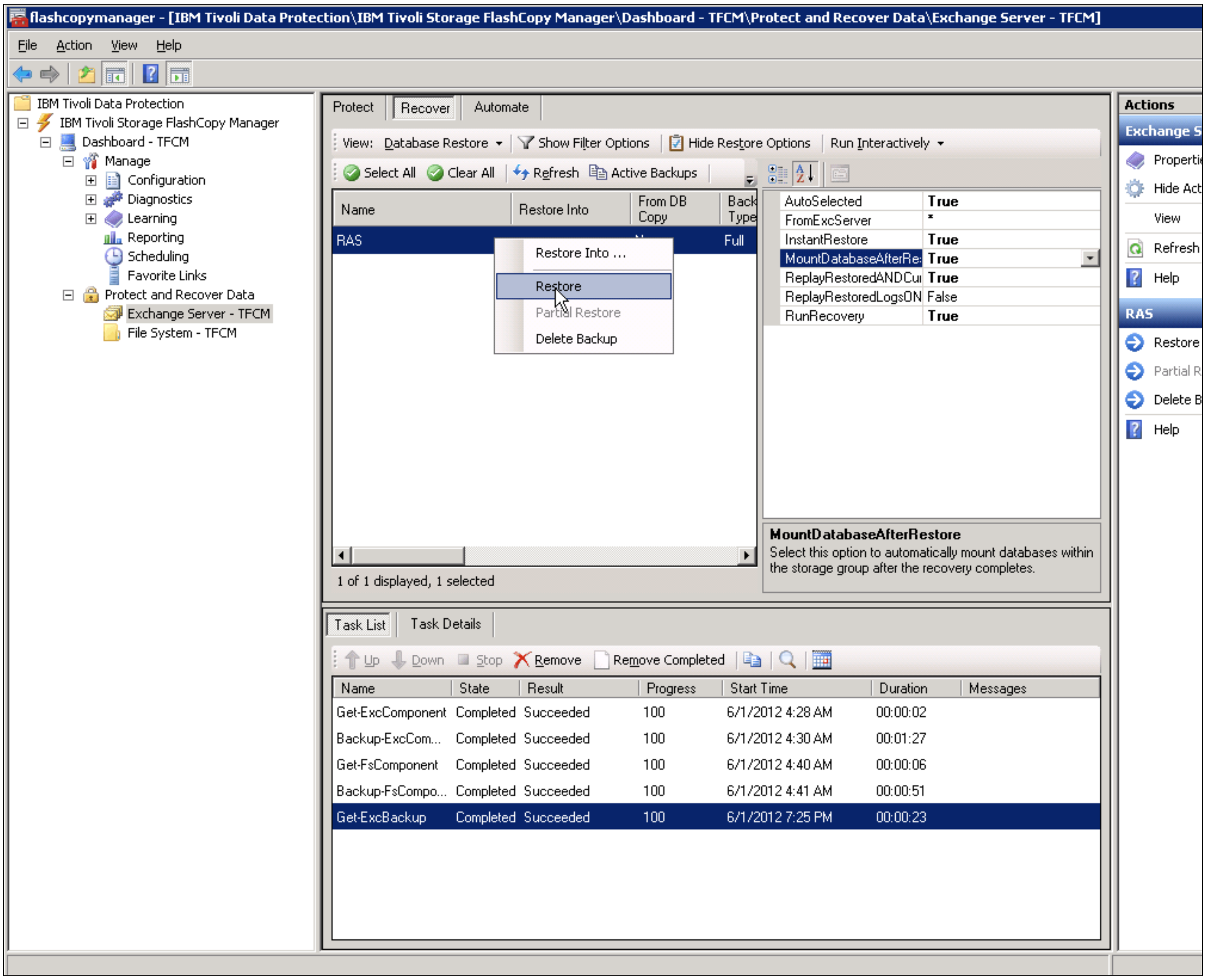


Figure 11-20 Exchange Database restore

If you check IBM Storwize V7000 operations at this precise moment, you see there are two FlashCopy operations from Target to Source Volumes (Figure 11-21).

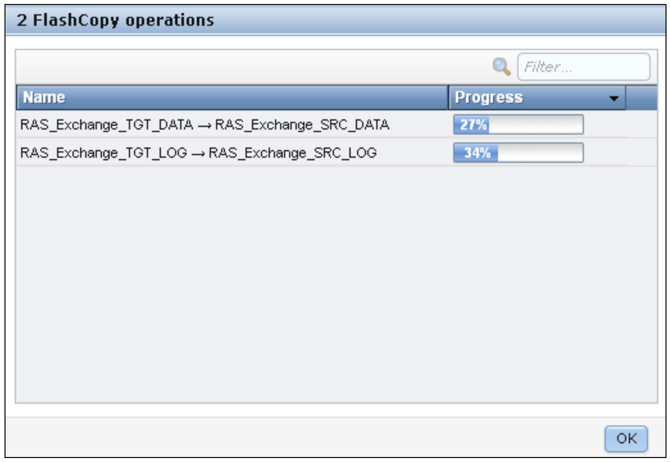


Figure 11-21 IBM Storwize V7000 operations during restore

The Exchange Database restore finished in 4:56 minutes (Figure 11-22).

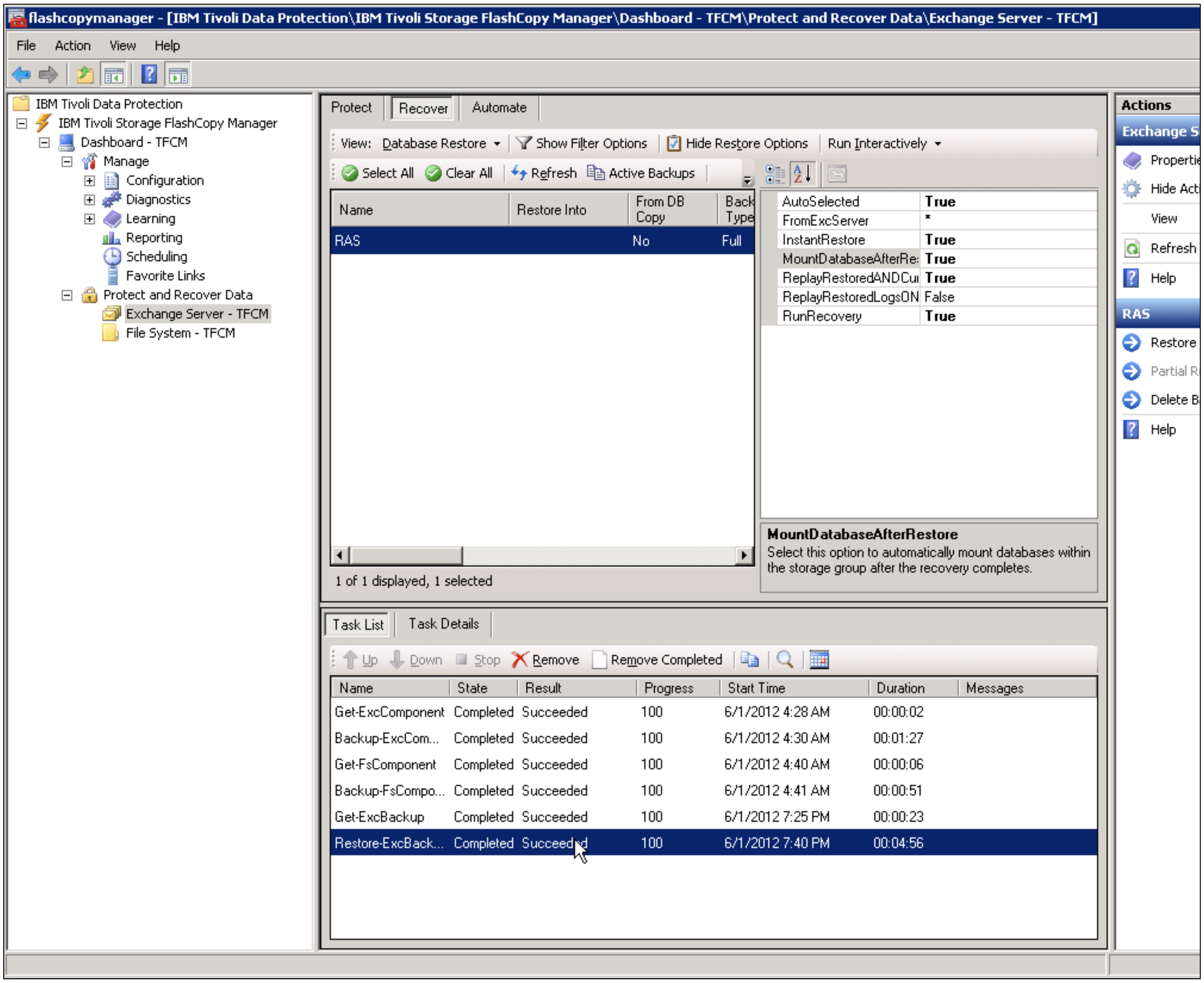


Figure 11-22 Exchange Database restore finished

Bob's mail box shows that all information is restored (Figure 11-23 on page 454).

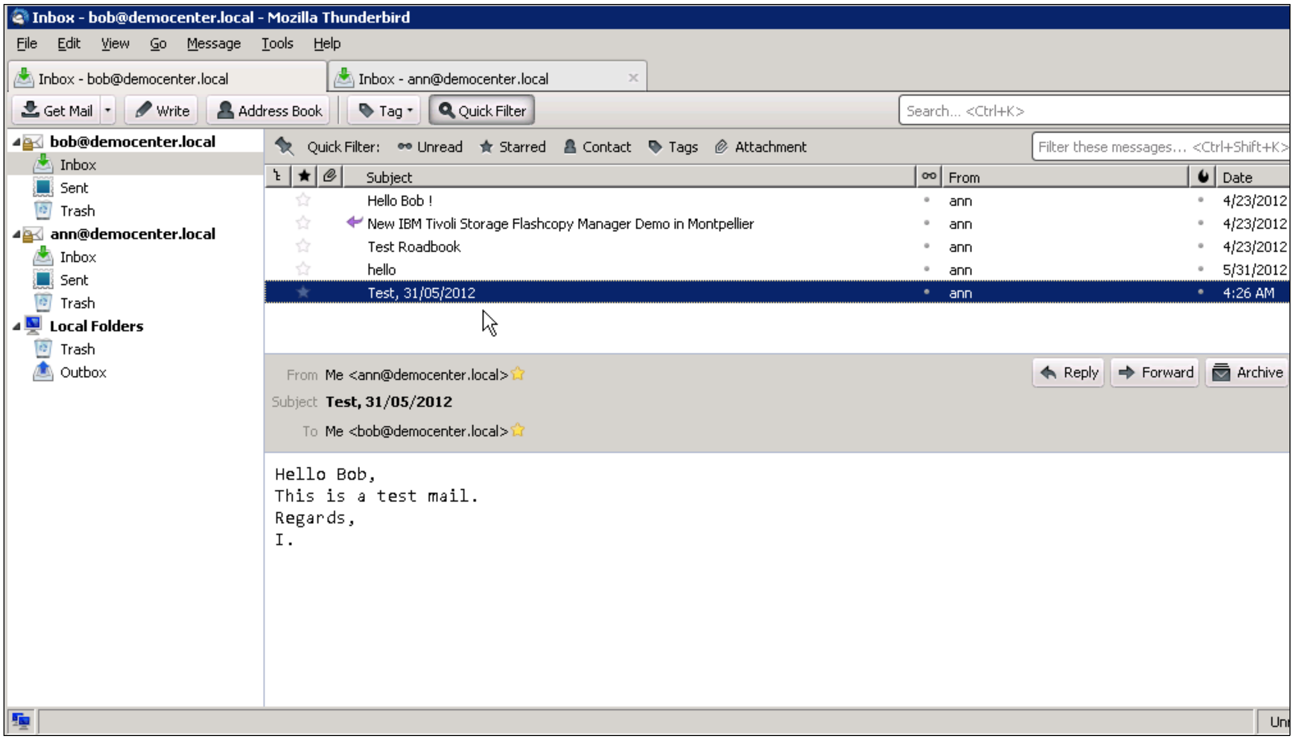


Figure 11-23 Exchange Database data restored

In cases with a large Exchange Database and a request to restore a particular mail box only, Tivoli Storage FlashCopy Manager provides another useful option to restore simply that mail box. From the Database Restore menu, select **Mailbox Restore** (Figure 11-24 on page 455).

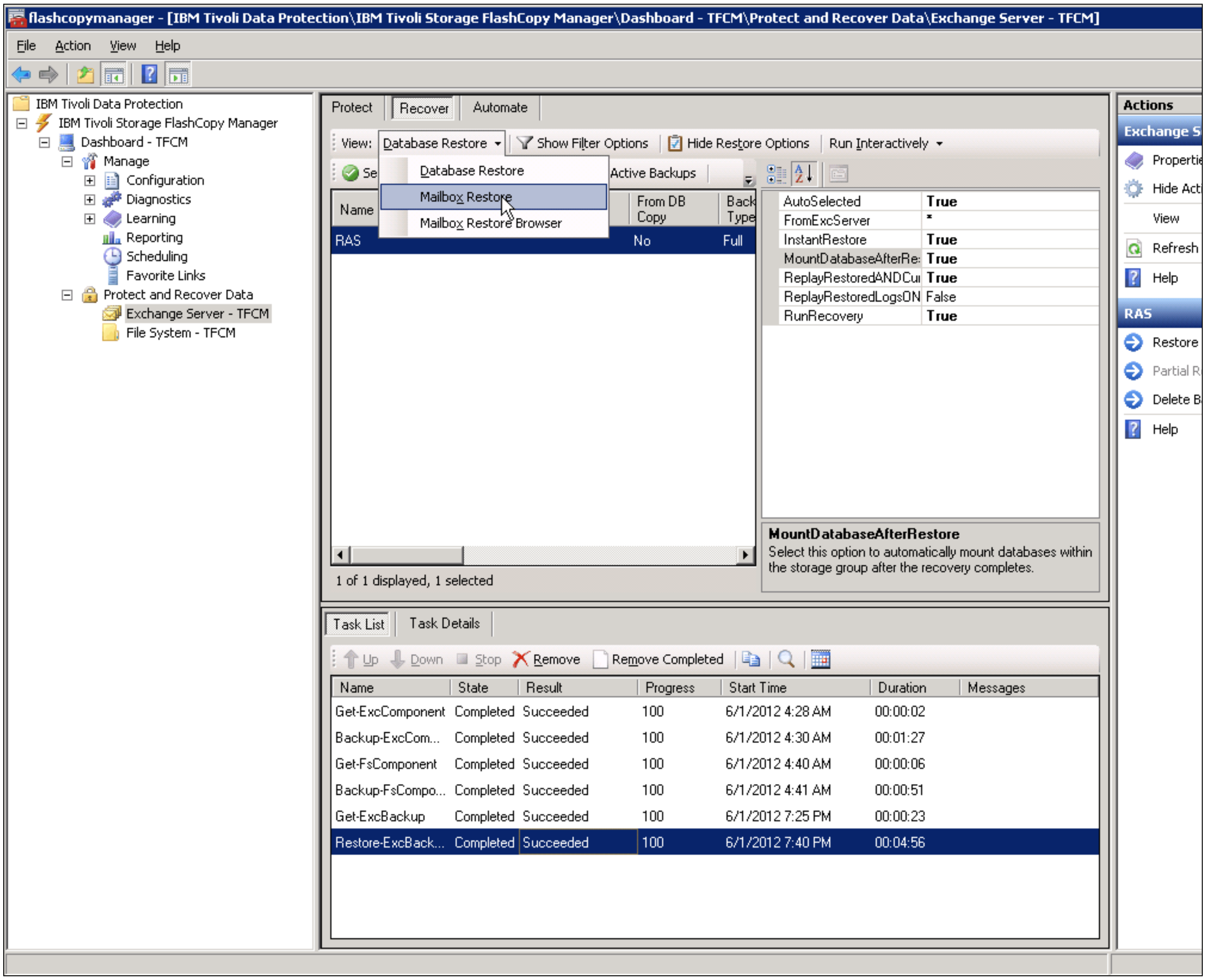


Figure 11-24 Mailbox Restore

Choose the desired Mailbox and select **Restore Mail to Original Location** (Figure 11-25 on page 456).

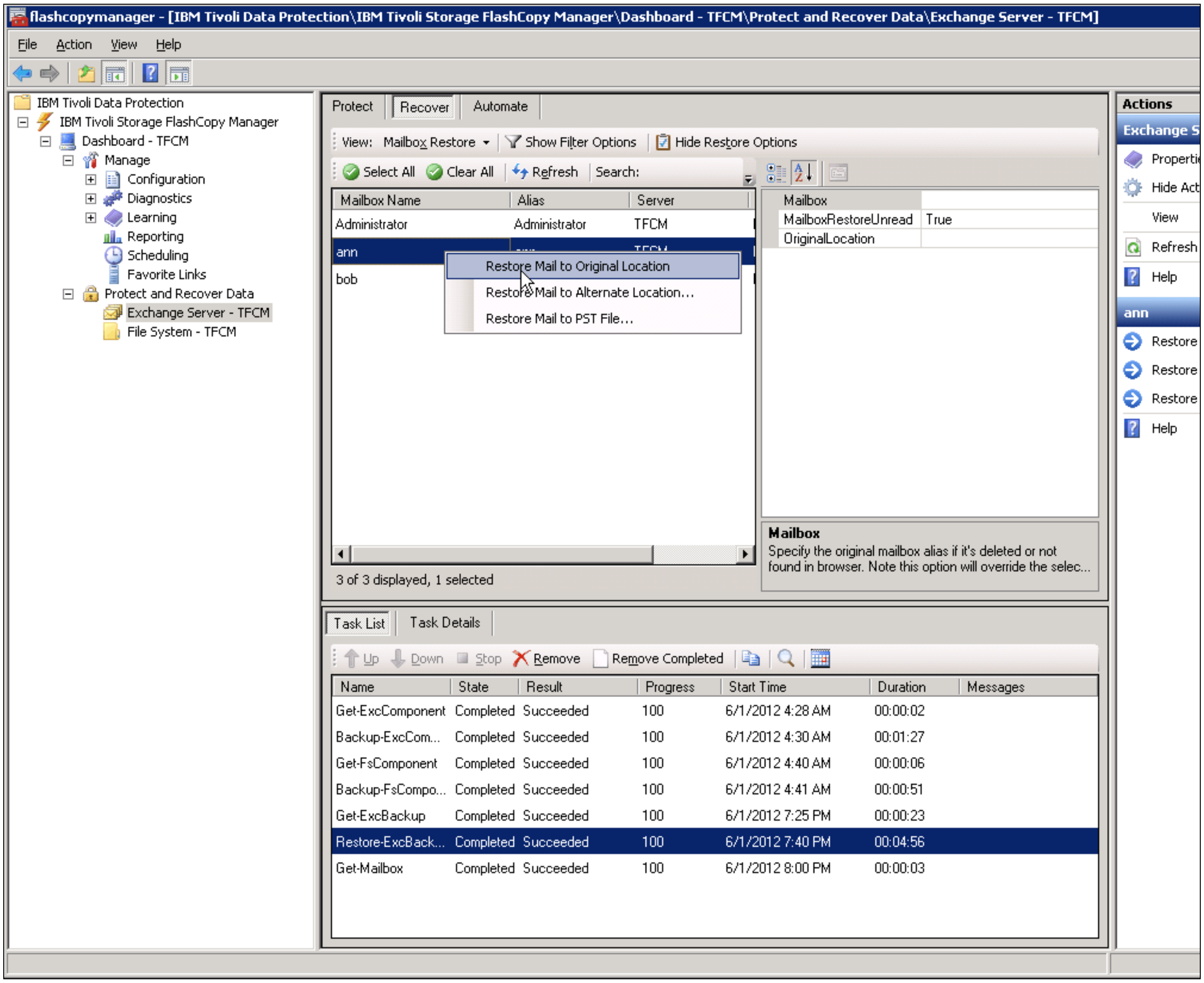


Figure 11-25 Choose desired Mailbox to restore

That particular Mailbox data has been restored. You can obtain a task list and status from the Tivoli Storage FlashCopy Manager Dashboard as shown in Figure 11-26 on page 457.

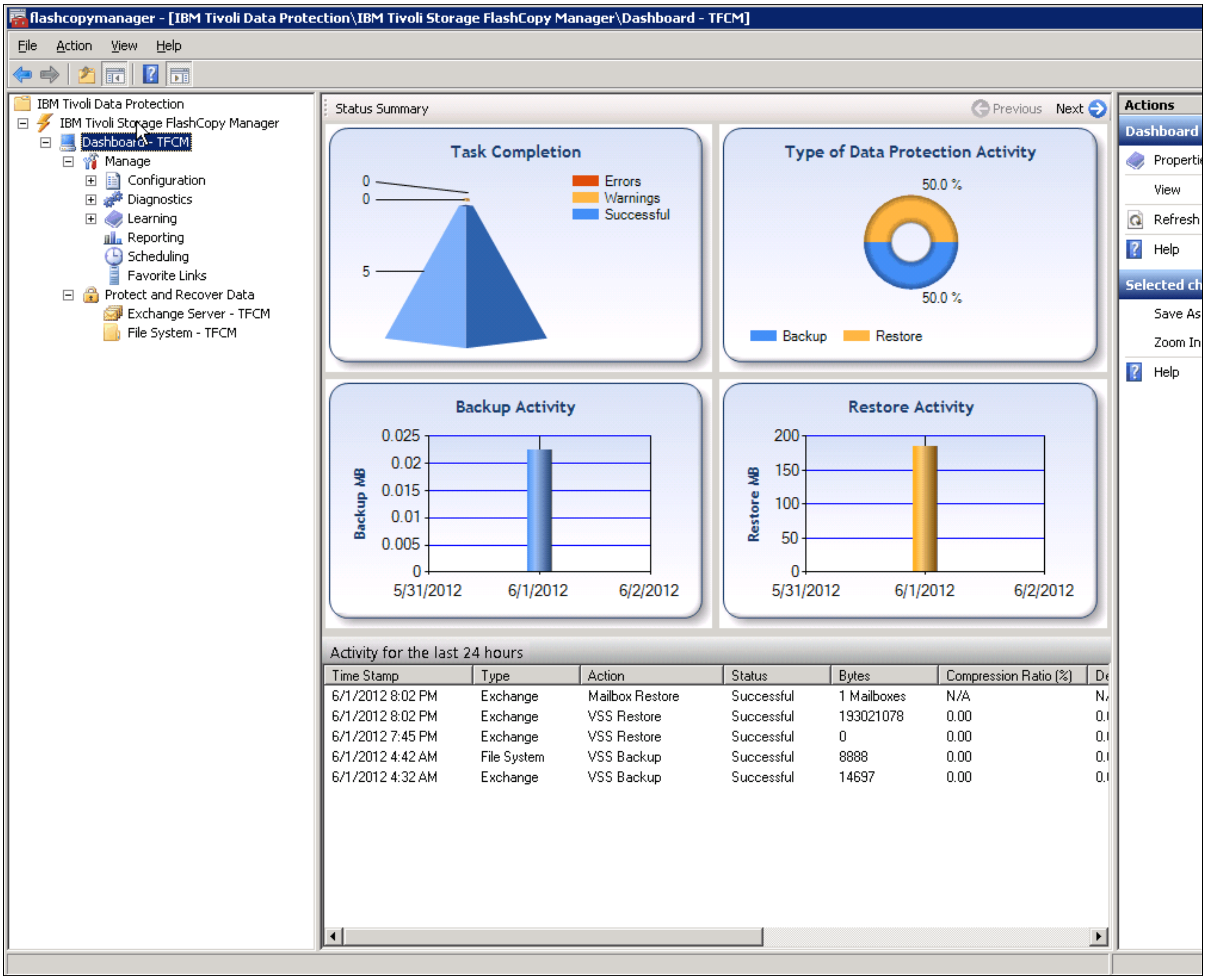


Figure 11-26 Task list and status

These two examples show a quick and easy way to protect your Microsoft Exchange environment using IBM Tivoli Storage FlashCopy Manager and IBM SVC/Storwize V7000 FlashCopy Services.

11.2.3 Microsoft File System backup and restore with IBM Tivoli Storage FlashCopy Manager for Windows

The following example shows a File System backup and restore with Tivoli Storage FlashCopy Manager. We have a Windows server with IBM SVC/Storwize V7000 Volume with an NTFS file system, Drive Letter W: and a size of 2 GB (Figure 11-27 on page 458).

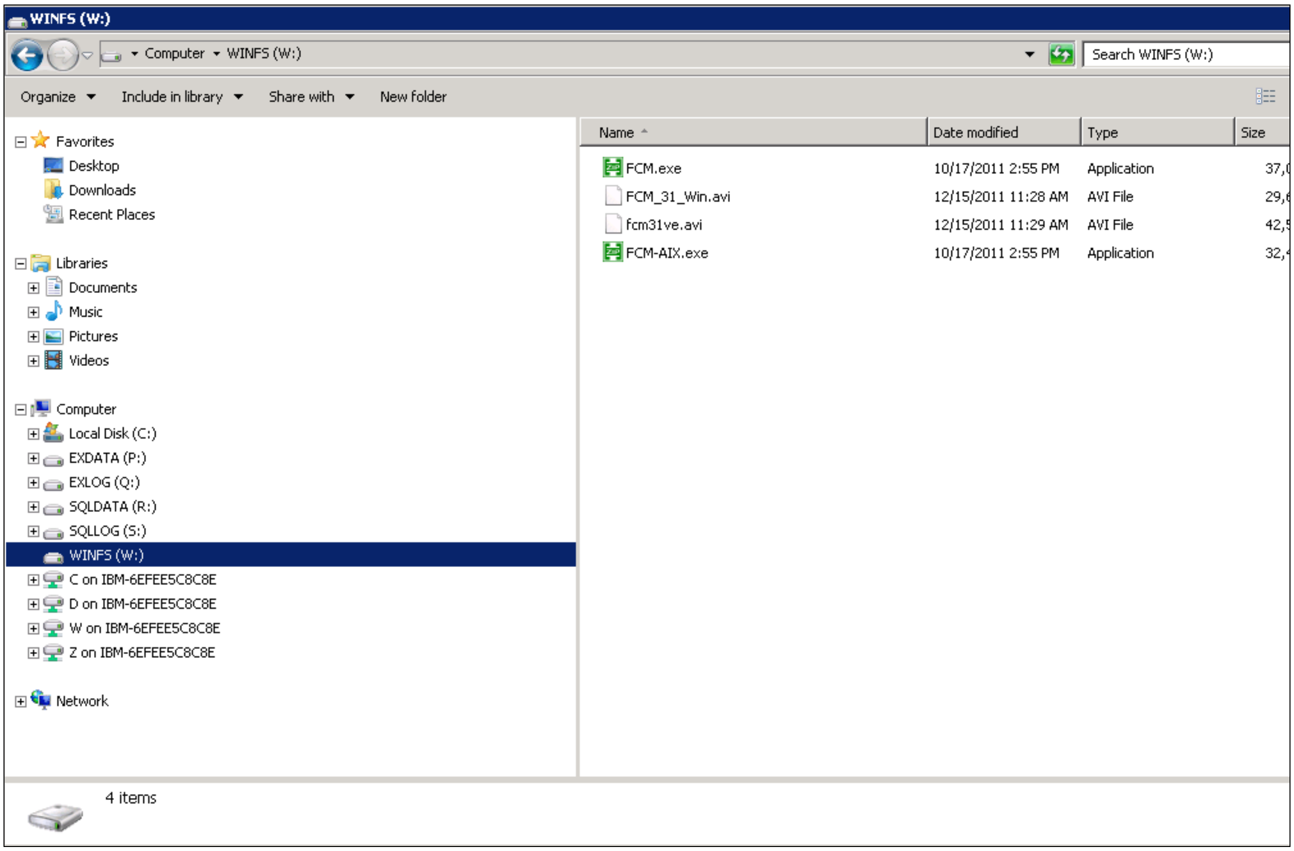


Figure 11-27 Windows file system

Using the Tivoli Storage FlashCopy Manager console, we create a full drive backup (Figure 11-28 on page 459).

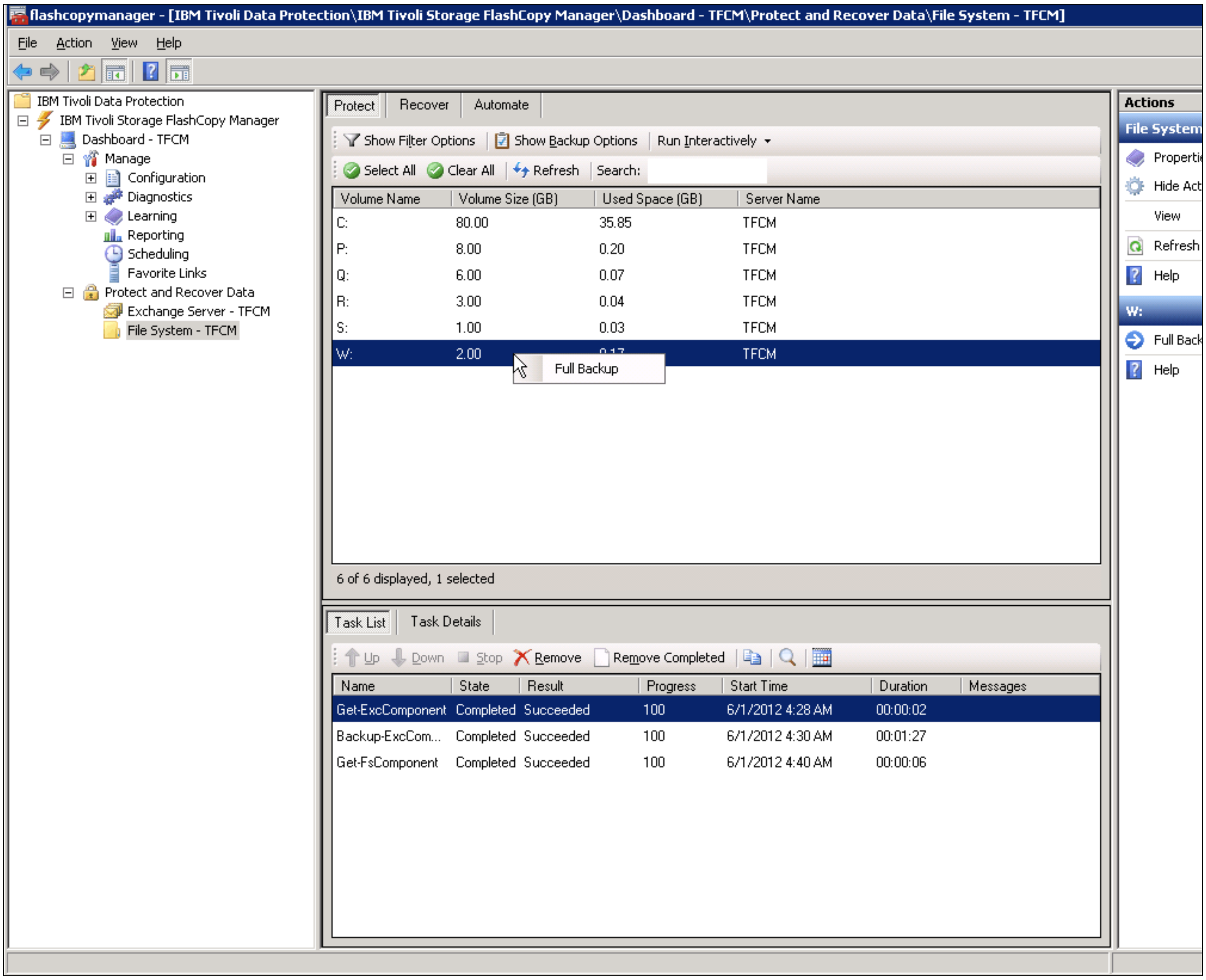


Figure 11-28 Full file system backup

Using the **ibmvcfg.exe** utility we check the status (Example 11-4).

Example 11-4 FlashCopy status

```
c:\Program Files\IBM\Hardware Provider for VSS-VDS>ibmvcfg list infc
Physical host.
Provider Type is SVC Pegasus 6.3.
Listing flashcopy relationship(s)...

                                        <Please use the target vol. name to del fc map>
FC ID FC Name         Src ID  Src Name        Tgt ID  Tgt Name        InFC    Cp.Rate Cp%  Cl.Rate Cl%
----------------------------------------------------------------------------------------------------------
0     fcmap3          23      RAS_WINFS_SRC   24      RAS_WINFS_TGT   false   50      100  50    100

Total: 1 pair(s) of flashcopy relationship.

c:\Program Files\IBM\Hardware Provider for VSS-VDS>
```

Then we delete several files (Figure 11-29 on page 460).

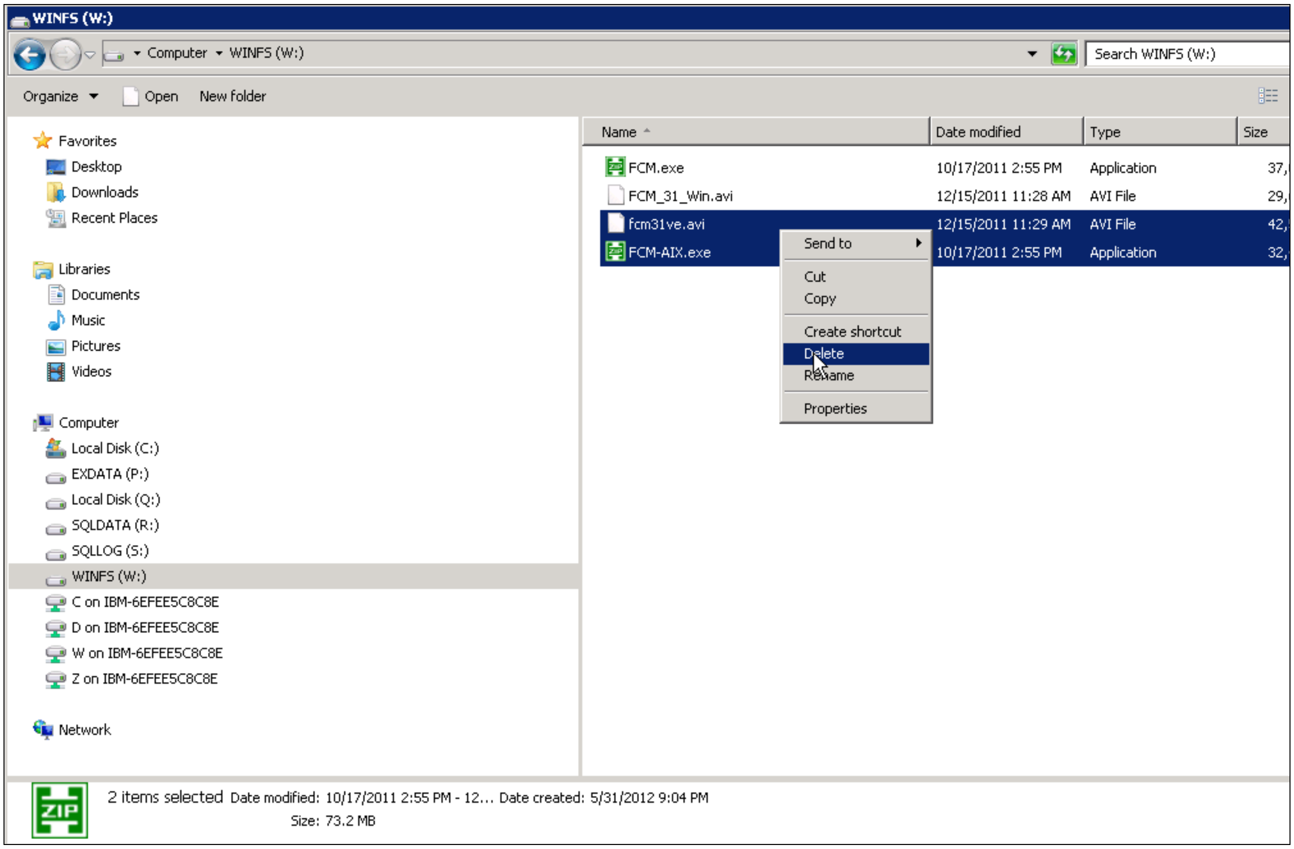


Figure 11-29 Delete several files

Next, we initiate a restore (Figure 11-30 on page 461).

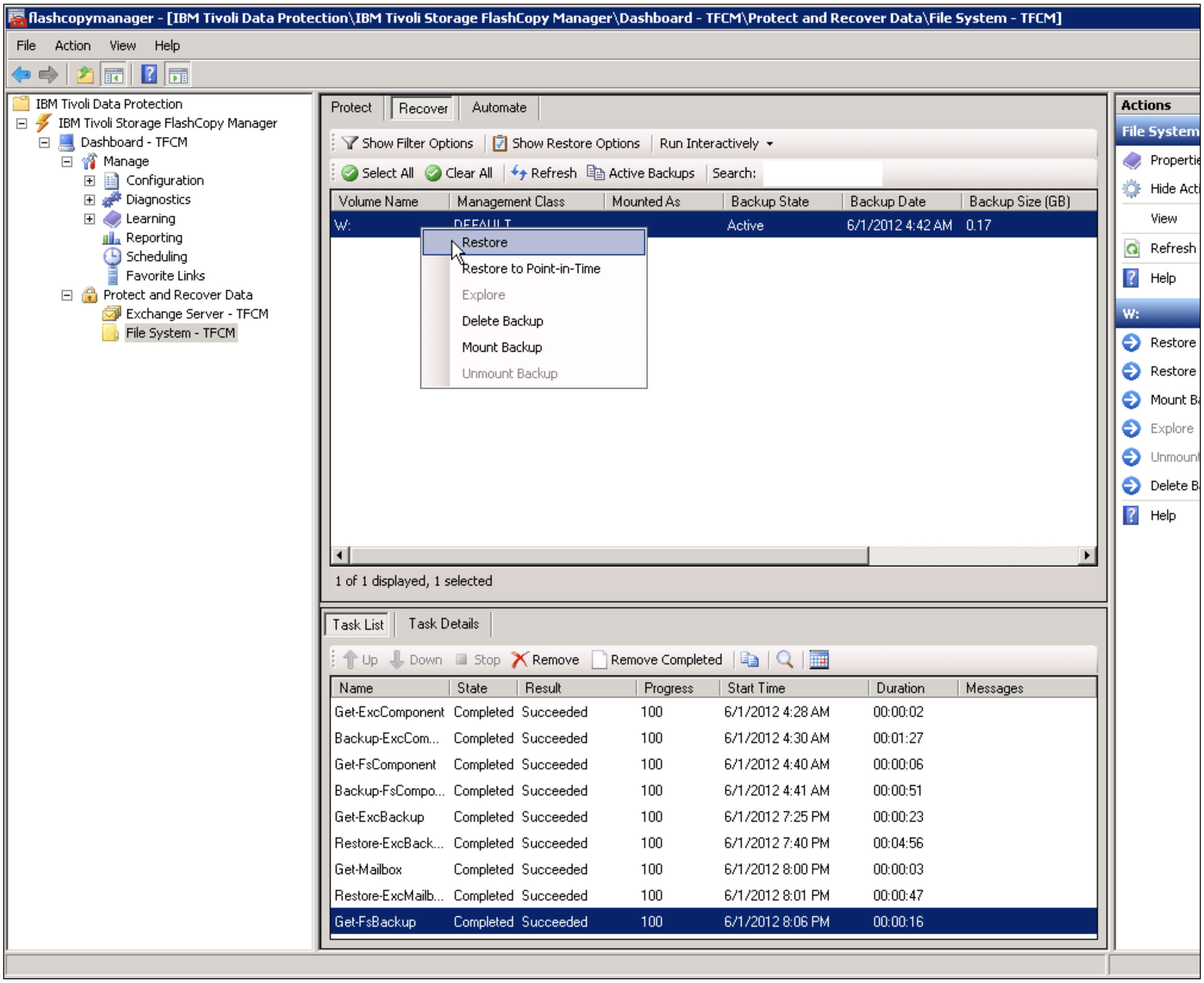


Figure 11-30 File System restore

Deleted files are quickly and easy restored as shown in Figure 11-31 on page 462.

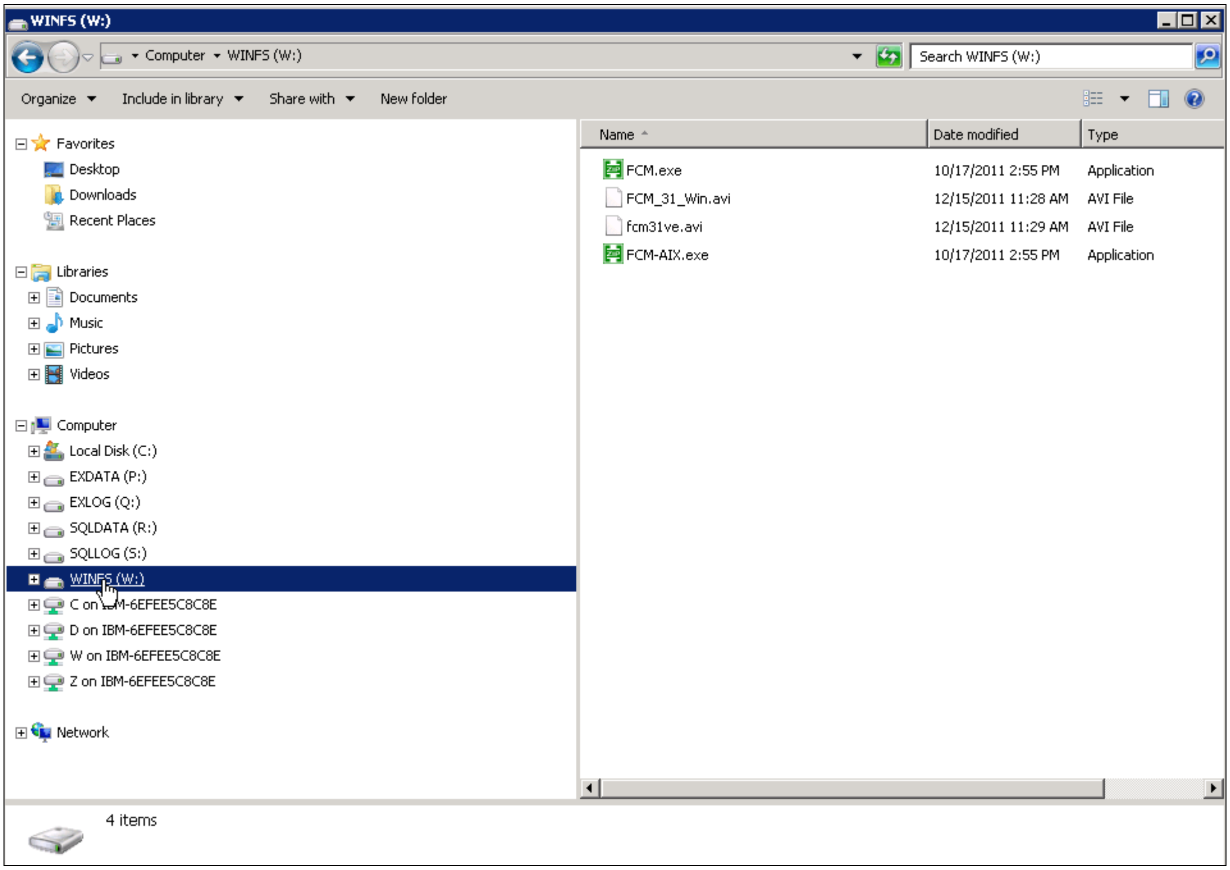


Figure 11-31 Files restored

For in-depth details about setup and configuration, refer to IBM Tivoli Storage FlashCopy Manager documentation.

11.2.4 IBM Tivoli Storage FlashCopy Manager for Oracle

Tivoli Storage FlashCopy Manager provides backup and recovery for native Oracle databases that creates full database backups and restores. Also database control files can be backed up into the Tivoli Storage FlashCopy Manager backup repository. Tivoli Storage FlashCopy Manager supports databases on file systems that are supported by Tivoli Storage FlashCopy Manager for your platform and on Oracle ASM running on raw devices.

In Example 11-5 we have Red Hat Enterprise Linux, a native Oracle Database and a Linux Ext3 file system. First we check the current managed capacity with the **fmquery** command to see the amount of storage space used for backups, and list of all source volumes protected by IBM Tivoli Storage FlashCopy Manager for which a backup has been created, as shown.

Example 11-5 Tivoli Storage FlashCopy Manager managed capacity before backup

```
[oracle@fcmoracle ~]$ cat /etc/redhat-release
Red Hat Enterprise Linux Server release 5.7 (Tikanga)
[oracle@fcmoracle ~]$ cd $ORACLE_HOME/acs
[oracle@fcmoracle acs]$ pwd
/home/oracle/product/11.2.0/acs
[oracle@fcmoracle acs]$ ./fmquery -p profile
```

```
IBM Tivoli Storage FlashCopy(R) Manager
managed capacity overview (fmquery) 3.1.0.0 (Build 536) compiled Sep 30 2011

FMM6246I Total managed capacity: 0 Bytes
[oracle@fcmoracle acs]$
```

We have not made any backups yet so the managed capacity is 0 Bytes. The application and database is up and running and we start IBM Tivoli Storage FlashCopy Manager backup as shown in Example 11-6.

Example 11-6 Tivoli Storage FlashCopy Manager backup

```
[oracle@fcmoracle acs]$ pwd
/home/oracle/product/11.2.0/acs
[oracle@fcmoracle acs]$ ./acsora -f backup

                IBM Tivoli Storage FlashCopy(R) Manager for Oracle
               - Version 3, Release 1, Level 0.0  for Linux x86_64 -
                      Build: 536 generated on Sep 30 2011
           (c) Copyright IBM Corporation, 2000, 2011, All Rights Reserved.

FMM0005I Start of program at: Fri 01 Jun 2012 02:42:50 AM CEST.
FMM8800I The command is: backup
FMM6201I Checking status of database.
FMM6225I Create database parameter file
'/home/oracle/product/11.2.0/dbs/initPROD.ora_fromSPfile' from SPfile.
FMM6223I Detected control file: /oralog/PROD/control01.ctl
FMM6223I Detected control file: /oralog/PROD/control02.ctl
FMM6223I Detected control file: /oralog/PROD/control03.ctl
FMM6224I Create control file copy:
/home/oracle/diag/rdbms/prod/PROD/trace/cntrl_copy_A0H2WJ415W
FMM1553I RMAN is logging to
/home/oracle/product/11.2.0/acs/logs/rman.PROD.20120601024250.log

FMM1510I New connection received from host fcmoracle.
FMM1514I *****> Device client connected.
FMM6219I Backup to Tivoli Storage Manager: NO
FMM1582I The target set VOLUMES_SET_1 will be used for the current backup.
FMM6901I Response to Init request.
FMM6902I Response to Partition request.
FMM4184I CIM Agent version for SVC: '6.3.0'.
FMM6903I Response to Prepare Flash request.
FMM0357I FlashCopy type is 'NOCOPY'.
FMM6230I Set table space files in backup mode.
FMM6212I Suspend database.
FMM6905I Response to Flash request.
FMM0589I Flushing the buffers to disk...
FMM0607I Freezing filesystem : /oradata.
FMM4183I Performing NOCOPY FlashCopy of source volume DemoFCMOracle_DatSrc to
target volume DemoFCMOracle_DatTgt1.
FMM0142I Snapshot started ...
FMM0143I Snapshot successful.
FMM0608I Thawing filesystem : /oradata.
FMM6214I Resume database.
FMM6231I End backup mode for table space files.
FMM6906I Response to Verify request.
```

```
FMM6959I Script 'LD_LIBRARY_PATH=/home/oracle/product/11.2.0/acs/xpyv/lib
/home/oracle/product/11.2.0/acs/fmcima -l /home/oracle/product/11.2.0/acs -N
DEVICE_CLASS:STANDARD.0 -T 20120601024304 -f backup' returned with code 0.

#SAVED ORCL__A0H2WJ415W /oradata/PROD/sysaux01.dbf

#SAVED ORCL__A0H2WJ415W /oradata/PROD/soe.dbf

#SAVED ORCL__A0H2WJ415W /oradata/PROD/system01.dbf

#SAVED ORCL__A0H2WJ415W /oradata/PROD/undotbs01.dbf

#SAVED ORCL__A0H2WJ415W /oradata/PROD/soeindex.dbf

#SAVED ORCL__A0H2WJ415W /oradata/PROD/temp01.dbf

#SAVED ORCL__A0H2WJ415W /oradata/PROD/users01.dbf
FMM6217I Database switched to next logfile.
FMM0020I End of program at: Fri 01 Jun 2012 02:46:18 AM CEST.
FMM0021I Elapsed time: 03 min 28 sec.
FMM0024I Return code is: 0.
[oracle@fcmoracle acs]$
```

This application is working all the time. We notice a small performance slowdown during the backup, but at the end it is back to normal. If you check IBM SVC/Storwize V7000, you will see the **Source** to **Target** Volume 100% Copied (Figure 11-32).

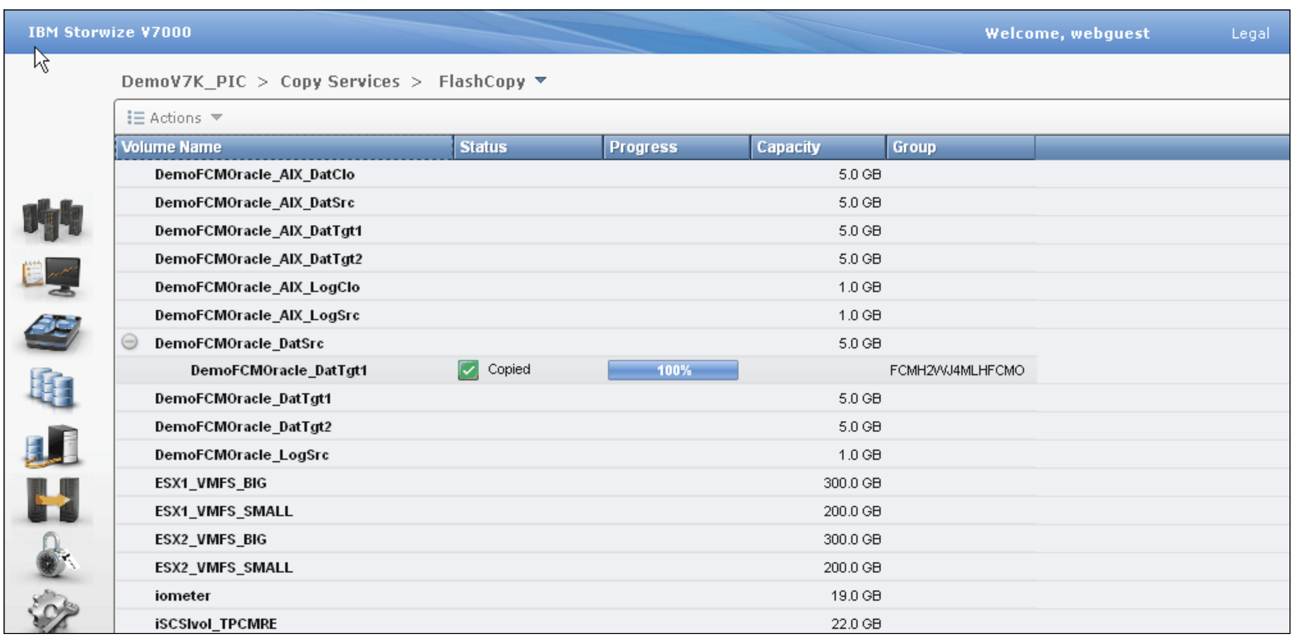


Figure 11-32 Storage Volume synchronization Source to Target

We check the managed capacity as shown in Example 11-7.

We have a Source Volume ID now and the total managed capacity 5 GB, which is exactly the size of the Source Volume.

Example 11-7 Tivoli Storage FlashCopy Manager managed capacity after backup

```
[oracle@fcmoracle acs]$ ./fmquery -p profile
IBM Tivoli Storage FlashCopy(R) Manager
managed capacity overview (fmquery) 3.1.0.0 (Build 536) compiled Sep 30 2011

FMM6245I source volume ID: 6005076801848375F00000000000172E  capacity: 5.000 GB
FMM6246I Total managed capacity: 5.000 GB
[oracle@fcmoracle acs]$
```

To perform the Oracle database restore we need to stop the Oracle database first using the scripts and techniques implemented during Oracle installation and setup. Then we run Tivoli Storage FlashCopy Manager restore as shown in Example 11-8.

Example 11-8 Tivoli Storage FlashCopy Manager restore

```
[oracle@fcmoracle acs]$ ./acsora -f restore

                IBM Tivoli Storage FlashCopy(R) Manager for Oracle
               - Version 3, Release 1, Level 0.0  for Linux x86_64 -
                     Build: 536 generated on Sep 30 2011
           (c) Copyright IBM Corporation, 2000, 2011, All Rights Reserved.

FMM0005I Start of program at: Fri 01 Jun 2012 08:54:43 PM CEST.
FMM8800I The command is: restore
FMM6232I Looking for the latest backup.
FMM6233I Restoring backup with ID ORCL__A0H2WJ415W.

FMM1510I New connection received from host fcmoracle.
FMM1514I *****> Device client connected.
FMM6901I Response to Init request.
FMM6904I Response to Restore request.
FMM4184I CIM Agent version for SVC: '6.3.0'.
FMM0511I ====>Performing IBM Tivoli Storage FlashCopy(R) Manager PREPARE SNAP
RESTORE command.
FMM0749I List of the current file systems on the backed up volume groups ...
oradatavg oradatalv /oradata
FMM0750I List of file systems which will be restored...
oradatalv /oradata  mount_options=rw,data=ordered
FMM0159I End of listing.
FMM0138I Time stamp: 06/01/12-20:54:48.
FMM6905I Response to Flash request.
FMM0511I ====>Performing IBM Tivoli Storage FlashCopy(R) Manager RESTORE command.
FMM0236I Disabling the volumes and filesystems ...
FMM0187I Creating a semaphore for the critical part of importing/exporting ...
FMM0188I Trying to set the semaphore for the critical part of importing/exporting
...
#UNMOUNTING_FS
FMM0172I Start of listing of exported volume groups/unmounting file systems ...
FMM0590I Unmounting the file system /oradata...
FMM0568I Removing volume group oradatavg ....
FMM0569I Varied off and exported volume group : oradatavg
FMM0189I Semaphore released.
FMM0138I Time stamp: 06/01/12-20:54:50.
FMM4183I Performing NOCOPY FlashCopy of source volume DemoFCMOracle_DatTgt1 to
target volume DemoFCMOracle_DatSrc.
```

```
FMM0142I Snapshot started ...
FMM0143I Snapshot successful.
FMM0225I Enabling the volumes and filesystems ...
FMM0591I Bringing up the volume groups...
FMM0188I Trying to set the semaphore for the critical part of importing/exporting
...
FMM0158I Start of listing of importing volume groups/mounting file systems ...
FMM0565I Importing volume groups now...
FMM0566I Newly imported volume group: oradatavg
FMM0543I Mounting filesystem : /oradata.
FMM0189I Semaphore released.
FMM0138I Time stamp: 06/01/12-20:54:56.
#FS_MOUNTED
FMM6907I Response to Complete Restore request.
FMM6959I Script 'LD_LIBRARY_PATH=/home/oracle/product/11.2.0/acs/xpyv/lib
/home/oracle/product/11.2.0/acs/fmcima -l /home/oracle/product/11.2.0/acs -N 0 -T
20120601205443 -f restore' returned with code 0.

#RESTORED ORCL__A0H2WJ415W /oradata/PROD/sysaux01.dbf

#RESTORED ORCL__A0H2WJ415W /oradata/PROD/soe.dbf

#RESTORED ORCL__A0H2WJ415W /oradata/PROD/system01.dbf

#RESTORED ORCL__A0H2WJ415W /oradata/PROD/undotbs01.dbf

#RESTORED ORCL__A0H2WJ415W /oradata/PROD/soeindex.dbf

#RESTORED ORCL__A0H2WJ415W /oradata/PROD/temp01.dbf

#RESTORED ORCL__A0H2WJ415W /oradata/PROD/users01.dbf
FMM0020I End of program at: Fri 01 Jun 2012 08:54:56 PM CEST.
FMM0021I Elapsed time: 13 sec.
FMM0024I Return code is: 0.
[oracle@fcmoracle acs]$
```

On IBM SVC/Storwize V7000 storage we can see Target to Source Volume synchronization in progress (Figure 11-33).

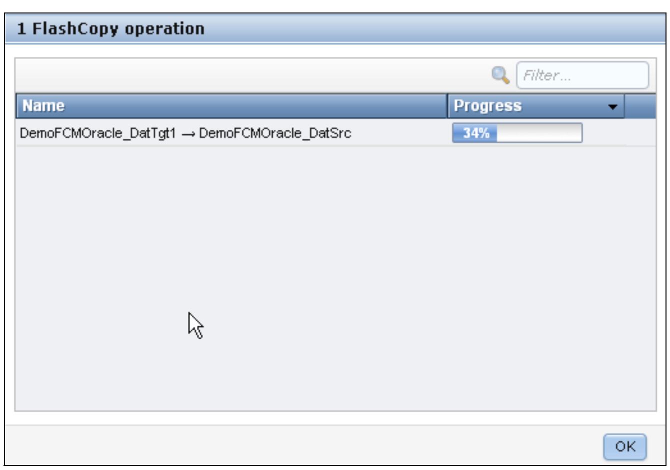


Figure 11-33 Storage Volume synchronization Target to Source

Next, we need to start the Oracle database as shown in Example 11-9. Key steps are database mount, recover and alter.

Example 11-9 Start Oracle database

```
SQL*Plus: Release 11.2.0.1.0 Production on Fri Jun 1 20:58:03 2012

Copyright (c) 1982, 2009, Oracle.  All rights reserved.

SQL> Connected to an idle instance.
SQL> ORACLE instance started.

Total System Global Area 1068937216 bytes
Fixed Size                  2220200 bytes
Variable Size             910167896 bytes
Database Buffers          150994944 bytes
Redo Buffers                5554176 bytes
Database mounted.
SQL> alter database recover
*
ERROR at line 1:
ORA-00279: change 1351763 generated at 06/01/2012 02:43:22 needed for thread 1
ORA-00289: suggestion : /home/oracle/archivelog/PROD/1_7_758714052.dbf
ORA-00280: change 1351763 for thread 1 is in sequence #7


SQL> alter database open
*
ERROR at line 1:
ORA-01108: file 1 is in backup or media recovery
ORA-01110: data file 1: '/oradata/PROD/system01.dbf'


SQL> Disconnected from Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

****************************************************************
Press <Return Key> to close this terminal
```

Using IBM Tivoli Storage FlashCopy Manager and IBM SVC/Storwize V7000 FlashCopy services, the Oracle database is quickly restored now and we have running application.

For details about setup and configuration, see the IBM Tivoli Storage FlashCopy Manager documentation.

11.2.5 IBM Tivoli Storage FlashCopy Manager for VMware

Tivoli Storage FlashCopy Manager for VMware protects the virtual infrastructure through automated data protection and recovery of your virtual machines using IBM SVC/Storwize V7000 Replication Family Services. Tivoli Storage FlashCopy Manager for VMware offers an easy-to-use interface that provides a way for you to manage the backup and recovery of virtual machines in a multiple VMware ESX server environment. With Tivoli Storage FlashCopy Manager for VMware, you can create off-host storage hardware snapshot backups from VMware virtual machines using IBM SVC FlashCopy service.

Tivoli Storage FlashCopy Manager for VMware provides two important features:

- Backup, restore, and disaster recovery operations for virtual machines are streamlined and simplified.
- File system-consistent backups are provided and the backup window of the virtual machine is reduced by using hardware snapshots of complete datastores in combination with offloaded backups to Tivoli Storage Manager.

An IBM Tivoli Storage FlashCopy Manager 3.1 for VMware demonstration can be found at:

http://www.youtube.com/watch?v=hhMHUI5Fhik

For details regarding installation, configuration, and operation refer to the IBM Info Center:

http://pic.dhe.ibm.com/infocenter/tsminfo/v6r3/index.jsp?topic=%2Fcom.ibm.itsm.nav.doc%2Ft_protect_fcmvmw.html

11.2.6 IBM Tivoli Storage FlashCopy Manager live demonstration

For Tivoli Storage FlashCopy Manager live demonstration refer to the IBM Virtual briefing Center at Montpellier, France:

http://ibm-vbc.centers.ihost.com/briefingcenter/montpellier

For a remote demonstration request refer to:

http://ibm-vbc.centers.ihost.com/schedule/request-briefing/?profile=1

For comprehensive, in-depth technology briefings, product demonstrations and solution workshops for IBM clients and IBM Business Partners wanting product expertise on IBM Systems and Storage solutions, refer to the IBM Executive Briefing Center:

http://www-03.ibm.com/systems/services/briefingcenter/

11.3 IBM SVC plug-ins

Several plug-ins have been developed to make IBM SVC/Storwize V7000 systems easy to manage and monitor, and to improve application performance by moving the load from server to storage hardware.

11.3.1 IBM Storage Management Console for VMware vCenter

VMware vCenter is a centralized management interface for VMware environment. IBM Storage Management Console for VMware vCenter is an easy way to grant VMware administrators access and management to their storage. This plug-in integrates in the VMware vCenter management console and enables you to create new volumes, resize existing volumes, delete old volumes, migrate volumes and map and unmap volumes.

After IBM Storage Management Console for VMware vCenter software is installed, an IBM Storage option will display in your vSphere management tab; IBM SVC/Storwize V7000 management is part of VMware management now as shown in Figure 11-34 on page 469.

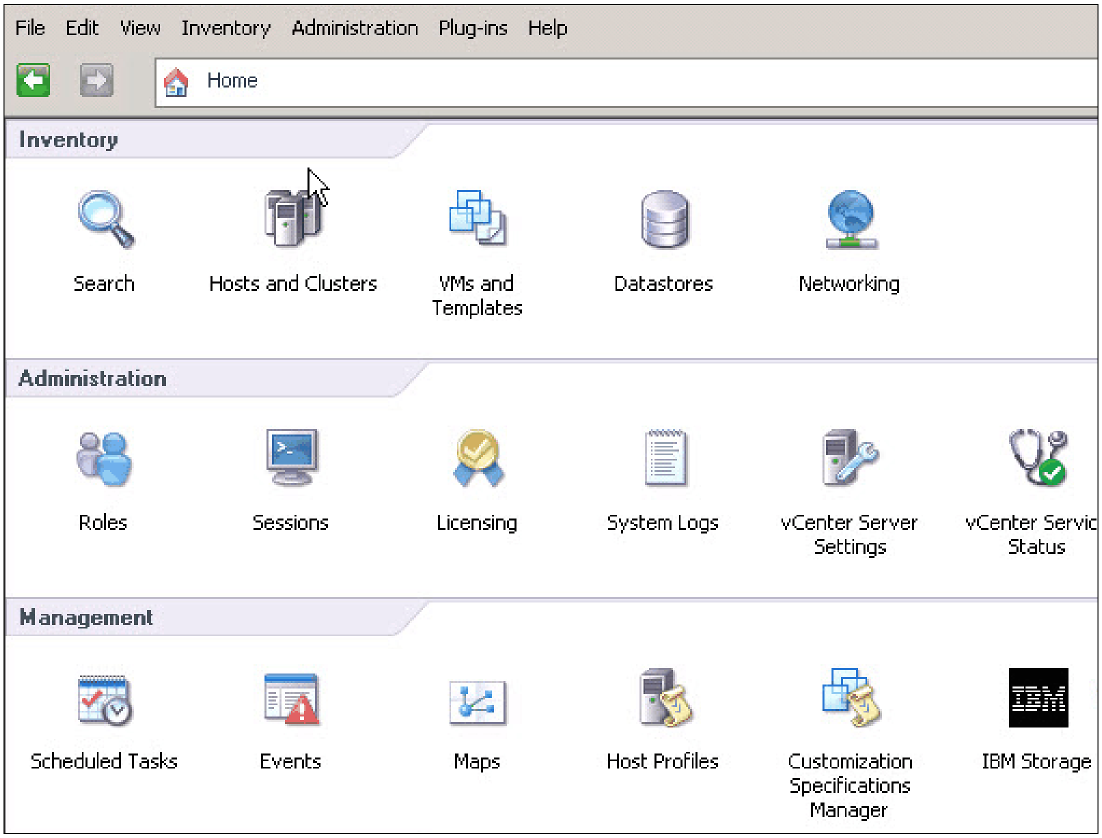


Figure 11-34 IBM Storage Management Console plug-in

For IBM SVC 5.1, 6.1, 6.2, IBM Storwize V7000 6.1, 6.2 and VMware vCenter 4.x (4.0, 4.1) you can use IBM Storage Management Console for VMware vCenter Version 2.5.1.

For IBM SVC 5.1, 6.1, 6.2, 6.3, IBM Storwize V7000 6.1, 6.2, 6.3, IBM Storwize V7000 Unified 1.3.1 and VMware vCenter 4.x (4.0, 4.1) or 5.0 you can use IBM Storage Management Console for VMware vCenter Version 3.0.0.

This plug-in quickly shows the relationship between virtual machines, VMware datastores, and IBM SVC/Storwize V7000 volumes, which is the first step of Remote Copy services setup. The plug-in is extremely useful for managing and supporting disaster recovery tasks and activities.

For installation, setup, and capability details, refer to IBM SVC documentation at:

http://publib.boulder.ibm.com/infocenter/svc/ic/index.jsp?topic=%2Fcom.ibm.storage.svc.console.doc%2Fmlt_relatedinfo_224agr.html

To download the software files, User Guide or Release Notes go to:

http://www-933.ibm.com/support/fixcentral/swg/selectFixes?parent=ibm/Storage_Disk&product=ibm/Storage_Disk/IBM+Storwize+V7000+%282076%29&release=All&platform=All&function=all

11.3.2 Storage Management Pack for Microsoft System Center Operations Manager

Microsoft System Center Operations Manager (SCOM) is a data center management and monitoring software for operating systems and hypervisors. The IBM Storage Management Pack for Microsoft System Center Operations Manager is a package set that enables you to integrate IBM SVC/Storwize V7000 systems monitoring and alerting in Microsoft SCOM console. This plug-in allows you to pick up alerts and events and obtain system status and graphs for your IBM Storage devices on a centralized monitoring console.

For installation, setup, and additional information, refer to IBM SVC documentation:

http://publib.boulder.ibm.com/infocenter/svc/ic/index.jsp?topic=%2Fcom.ibm.storage.svc.console.doc%2Fmlt_relatedinfo_224agr.html

11.3.3 IBM Storage Device Driver for VMware

VMware vStorage APIs for Array Integration (VAAI) offloads several storage-intensive tasks to the storage systems so that the ESX hosts are freed to perform other tasks.

IBM Storage Device Driver for VMware VAAI is a plug-in that allows VMware to offload the following storage block operations and tasks to IBM SVC/Storwize V7000 controllers:

- Volume cloning without read/write operations on VMware hosts
- Block zeroing on large storage areas during VM initiation operations
- Hardware-assisted locking of range of blocks in shared volumes

These features move load from virtual machines hardware to IBM SVC/Storwize V7000 systems, thereby improving VMware task performance.

The latest version of IBM Storage Device Driver for VMware VAAI Version 1.2.0 supports IBM SVC 6.2, IBM Storwize V7000 6.2 and VMware ESX 4.1 or ESXi 4.1.

For installation, setup and additional information, refer to the IBM Fix Central site:

http://www-933.ibm.com/support/fixcentral/

11.4 IBM SVC/Storwize V7000 disaster recovery solution for VMware environment

VMware is one of the most popular software packages for virtualization on x86 machines. Many clients are using VMware for their production server and desktop environments, so disaster recovery features on virtualized servers, storage, software, and hardware are quite important.

VMware Site Recovery Manager (SRM) and IBM SVC/Storwize V7000 automate disaster recovery process in VMware environment and avoid running manual commands and scripts. This can be critical in emergency situations and in meeting meet high availability business requirements. This is an efficient way to reduce RPO and RTO, and an easy way to plan and execute disaster recovery tests with VMware and IBM SVC/Storwize V7000 infrastructure.

For details about the implementation, see *Implementing disaster recovery solutions with IBM Storwize V7000 and VMware Site Recovery Manager* at (starts PDF download):

http://www-03.ibm.com/support/techdocs/atsmastr.nsf/5cb5ed706d254a8186256c71006d2e0a/906fb3333c0b35b0862577b400786956/$FILE/0003%20-%20Implementing%20DR%20solutions%20with%20IBM%20Storwize%20V7000%20and%20VMware%20Site%20Recovery%20Manager.pdf

Storage Replication Adapter (SRA) for VMware SRM is the link/adapter between VMware SRM and IBM SVC. It makes VMware software fully aware of IBM SVC/Storwize V7000 features such as Flash Copy and Remote Copy features. It gives SRM the possibility to automate recovery process using IBM SVC/Storwize V7000 storage Replication Family Services.

You can download the latest SRA software and installation instructions here:

ftp://ftp.software.ibm.com/storage/ds_open_api/VMWARE/SVC_SRA/

From a disaster recover perspective, if you use VMware, then IBM SVC/Storwize V7000 is a perfect storage choice. It is flexible and reliable with integrated Replication Family Services. To see IBM SVC/Storwize V7000 integration and features with VMware:

http://www.youtube.com/watch?v=SDZlghdzmoU

For another example of how to set up VMware with SAP and IBM Storwize V7000, visit the following site:

http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101913

11.5 IBM SVC/Storwize V7000 FlashCopy and Symantec NetBackup

Symantec NetBackup is a popular heterogeneous backup and recovery suite. It provides cross-platform backup functionality to a large variety of Windows, UNIX and Linux operating systems. Symantec NetBackup and IBM SVC/Storwize V7000 storage systems can be combined to greatly reduce or eliminate the need for backup windows or application downtime. A point-in-time copy of the client data is taken by using IBM SVC/Storwize V7000 FlashCopy function, and then you back up the data from the point-in-time copy using Symantec NetBackup.

For details regarding installation, setup, and scripting, visit *Symantec NetBackup with IBM Storwize V7000 FlashCopy or IBM XIV Snapshots*:

https://www-304.ibm.com/partnerworld/wps/servlet/download/DownloadServlet?id=hssXsLTLtZsiPCA$cnt&attachmentName=Symantec+NetBackup+with+IBM+Storwize+V7000+FlashCopy+or+IBM+XIV+Snapshots.pdf&token=MTMzNzkwNjg0MTUzNQ==&locale=en_ALL_ZZ

For a useful example of how to protect the Microsoft SQL database using Symantec NetBackup and IBM Storwize V7000 FlashCopy function, see the document *IBM Storwize V7000 SQL Server NetBackup VSS Solution*, which is available at the following site:

http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101763

For configuration and preferred practices for protecting Microsoft Exchange with Symantec NetBackup and IBM Storwize V7000, refer to *IBM Storwize V7000 Exchange Server 2010 NetBackup 7 VSS Solution*, which is available at the following site:

http://www-03.ibm.com/support/techdocs/atsmastr.nsf/5cb5ed706d254a8186256c71006d2e0a/92abe3255b212abd862577b500211f29/$FILE/IBM%20Storwize%20V7000%20Exchange%20Server%202010%20NetBackup7%20VSS%20solution%20v3.1%20updated.pdf

IBM SVC/Storwize V7000 FlashCopy function provides an efficient and cost-effective copy service to back up different applications and reduce the backup window with Symantec NetBackup software.

11.6 SVC/Storwize V7000 Remote Copy and Veritas Storage Foundation Enterprise HA

Symantec Veritas Storage Foundation is a storage solution for heterogeneous online storage management. Based on Veritas Volume Manager and Veritas File System, this storage solution provides a standard set of integrated tools to centrally manage data growth, maximize storage investments, provide data protection and flexibility.

Using IBM SVC/Storwize V7000 Metro Mirror and Global Mirror services, Veritas Storage Foundation, and Veritas Cluster Server, you can set up high availability and disaster recovery solutions for different enterprise applications.

For installation and configuration details refer to *Disaster recovery using Veritas Storage Foundation Enterprise HA with IBM Storwize V7000 Metro Mirror and Global Mirror* at:

http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101908

12

Fabric design considerations for Replication Family

A stable deployment of SVC replication must meet requirements beyond that of a normal SVC installation. A replication installation that does not meet these standards is more likely to disconnect during periods of high workload and cause a performance impact to your host.

The following topics are discussed:

- Topology and zoning for Global Mirror
- FCoE and remote replication
- Topology and zoning for Stretched Cluster
- WAN connectivity for replication
- SAN routing with mirroring

12.1 Topology and zoning for Metro and Global Mirror

The most intuitive zoning configuration for a mirroring installation is to simply create a “mirroring zone” that contains all ports from both clusters that you want to connect. Although such a configuration will work, field experience has shown that there are better ways to implement zoning.

The current zoning guidelines for mirroring installations advise that a maximum of two ports on each SVC node/V7000 canister be used for mirroring. The remaining two ports on the node/canister should not have any visibility to any other cluster. If you have been experiencing Node Asserts or cluster-wide performance issues when mirroring is in operation, implementing zoning in this fashion might help to alleviate this situation.

Each cluster-pair should have its own unique mirroring zone. Also, if two clusters are *not* in a copy relationship, they should *not* be visible to each other through zoning. Each cluster may be in a relationship with a maximum of three other clusters.

12.1.1 Dual WAN zoning

If you have two fully separate fabrics, zone your SVC/V7000 as shown in Figure 12-1 on page 475.

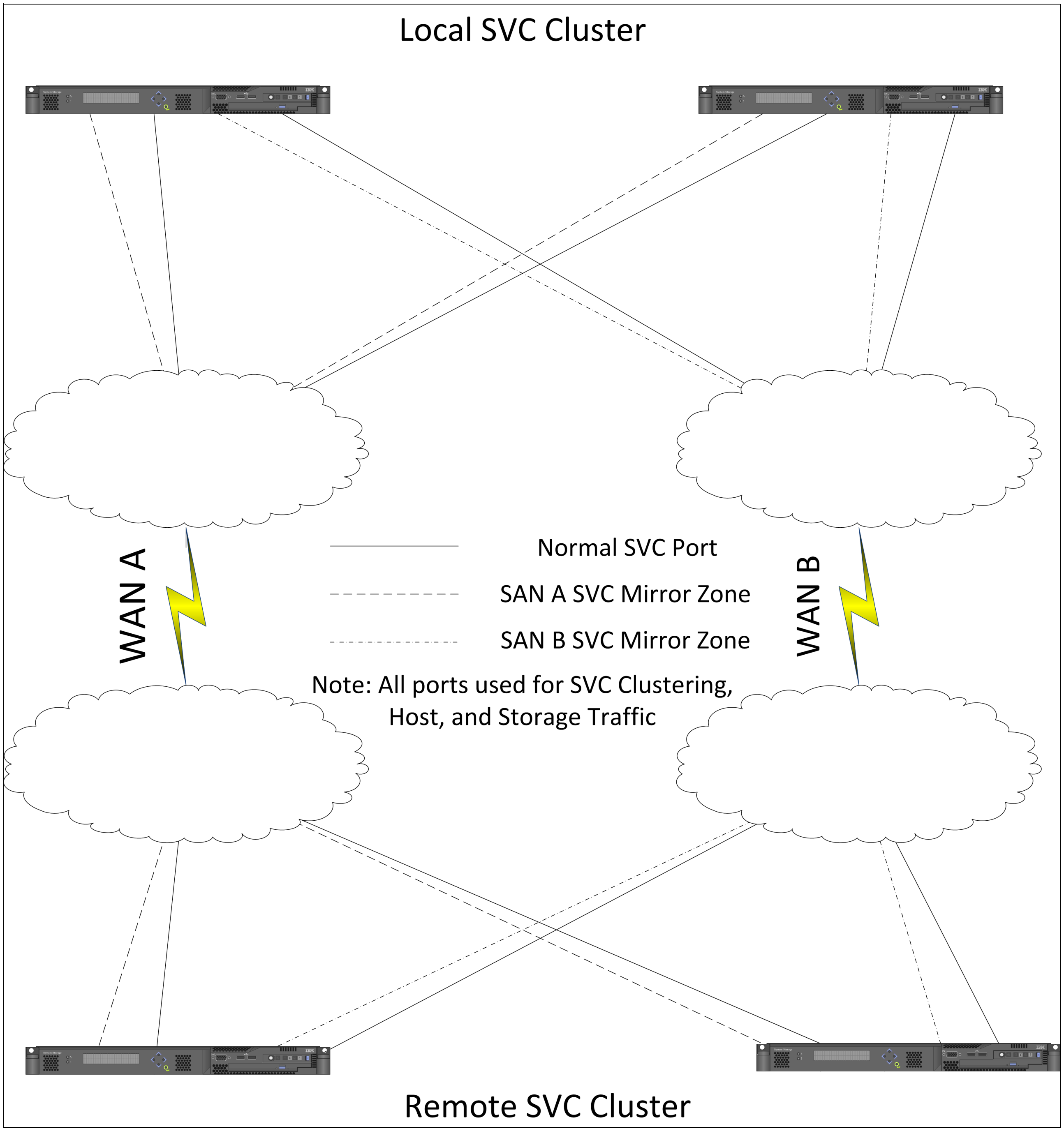


Figure 12-1 Dual WAN SVC mirror zoning

12.1.2 Single WAN zoning

If you only have a single pair of SAN routers providing the connection between your two sites, it is advisable to only use one fabric for replication instead of connecting both SAN fabrics to a single router. Note, though, that this does not apply if you are using something like a DWDM.

If you connect both fabrics to a single router, you eliminate much of the protection normally provided by having redundant SANs, because any sort of defect in the router software can potentially destabilize both SANs simultaneously.

12.1.3 Mirror-only ports

A new feature pending in SVC/V7000 V6.4 is the ability to zone ports (either FC or FCoE) only for mirroring traffic, and not use them for any other use, such as host, storage, or intra-cluster traffic. The new rules state that each SVC node or V7000 canister must have a minimum of two ports (either FC and/or FCoE) that have visibility to all other nodes or canisters in the cluster; the old rules required all four FC ports to have visibility to all other nodes or canisters. The remaining ports may be used for whatever purpose you want.

Only use this configuration, however, if you are confident that the ports you are not dedicating to mirroring can handle the full load of your regular I/O (including host, storage, and intra-cluster traffic).

To accomplish this, each pair of source/destination ports will need a unique zone. Thus, in any zone such as an SVC cluster zone or a mirroring zone, if more than one port in the same cluster is present, your two-port source/destination zones will be futile because that zone will then be used for intra-cluster SVC traffic.

> **Zoning support notice:** At the time of writing, this function is only supported on the FlexSystem V7000 and V7000 Unified. Support for SVC and standard V7000 is expected shortly.

12.1.4 Alternate Global Mirror topology - “forwarding” I/O group

If you are concerned about the effects of WAN congestion on the host I/O that shares the same I/O group as Global Mirror traffic, one thing that can lessen this potential impact is a “forwarding” I/O group.

This term means that, instead of having all of your local nodes communicate with all of your remote nodes, you choose one or more I/O groups on each end and only put those I/O groups in your GM zones. The SVC/V7000 will detect which nodes have inter-cluster links, and any GM I/O will be forwarded to those nodes before the I/O is sent to the other cluster. Figure 12-2 on page 477 illustrates this topology.

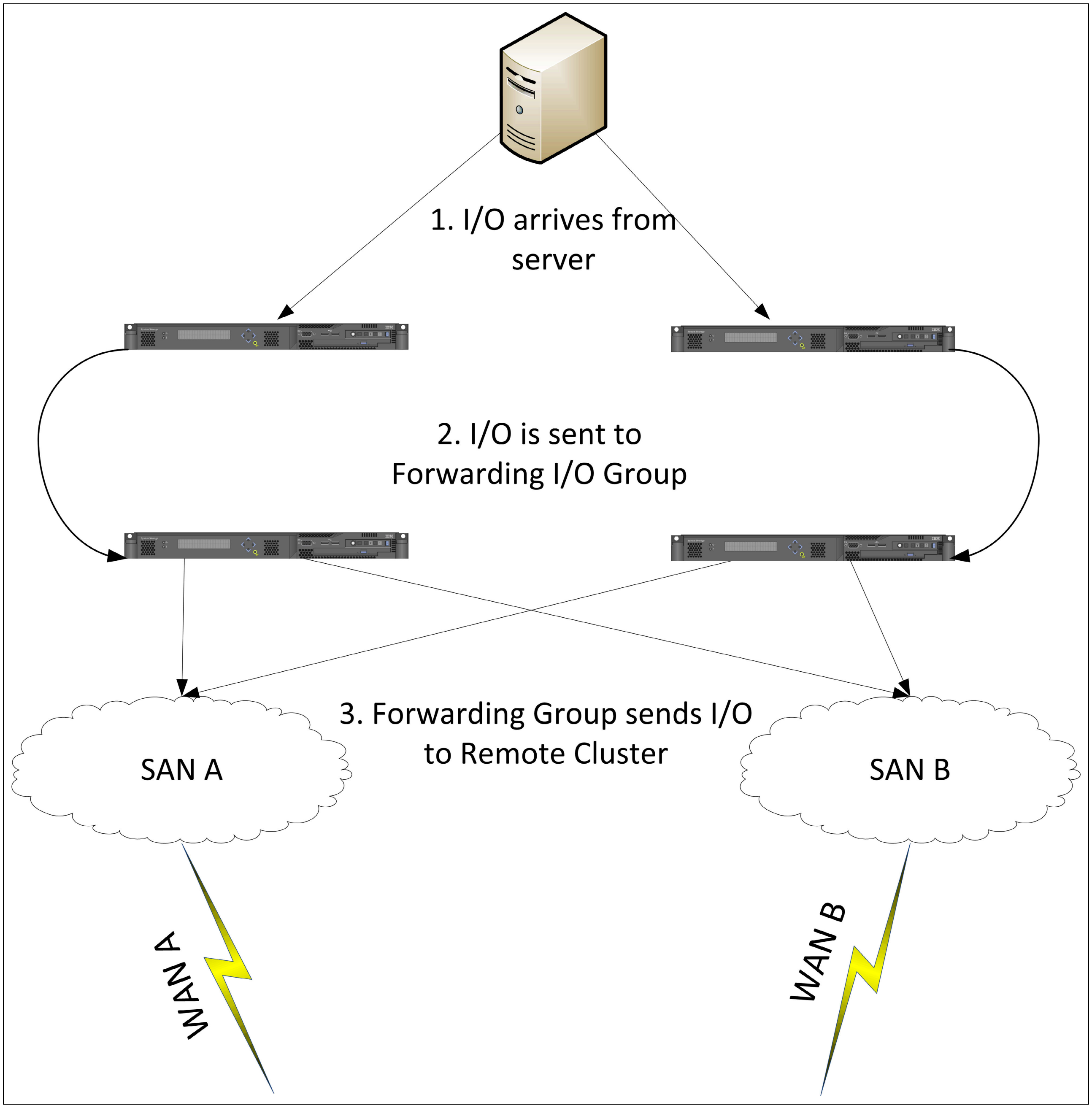


Figure 12-2 Forwarding I/O group traffic flow

In the event of WAN back pressure (and ensuing SAN buffer credit starvation), the effects this normally might have on primary I/O performance will be greatly reduced.

Because the SVC is licensed by software capabilities, as opposed to the hardware configuration, this can be a useful performance buffer for the cost of some SAN ports and an additional pair of SVC nodes. This assumes you are not currently running at a maximum cluster configuration already.

12.2 FCoE and remote replication

From the perspective of the SVC or V7000, anything a Fibre Channel port does may also be done with an FCoE port. However, there are minor limitations.

Current SVC/Storwize V7000 software does not support having more than one Fibre Channel Forwarder (FCF) be visible to a given FCoE port. Also, with current FCoE switch architectures, all FCoE traffic must go to and from the FCF. Direct traffic flow between two FCoE ports is not currently implemented if those ports are not directly connected to the same FCF. This is a FCoE limitation, not an SVC/V7000 limit.

This arrangement is illustrated in Figure 12-3.

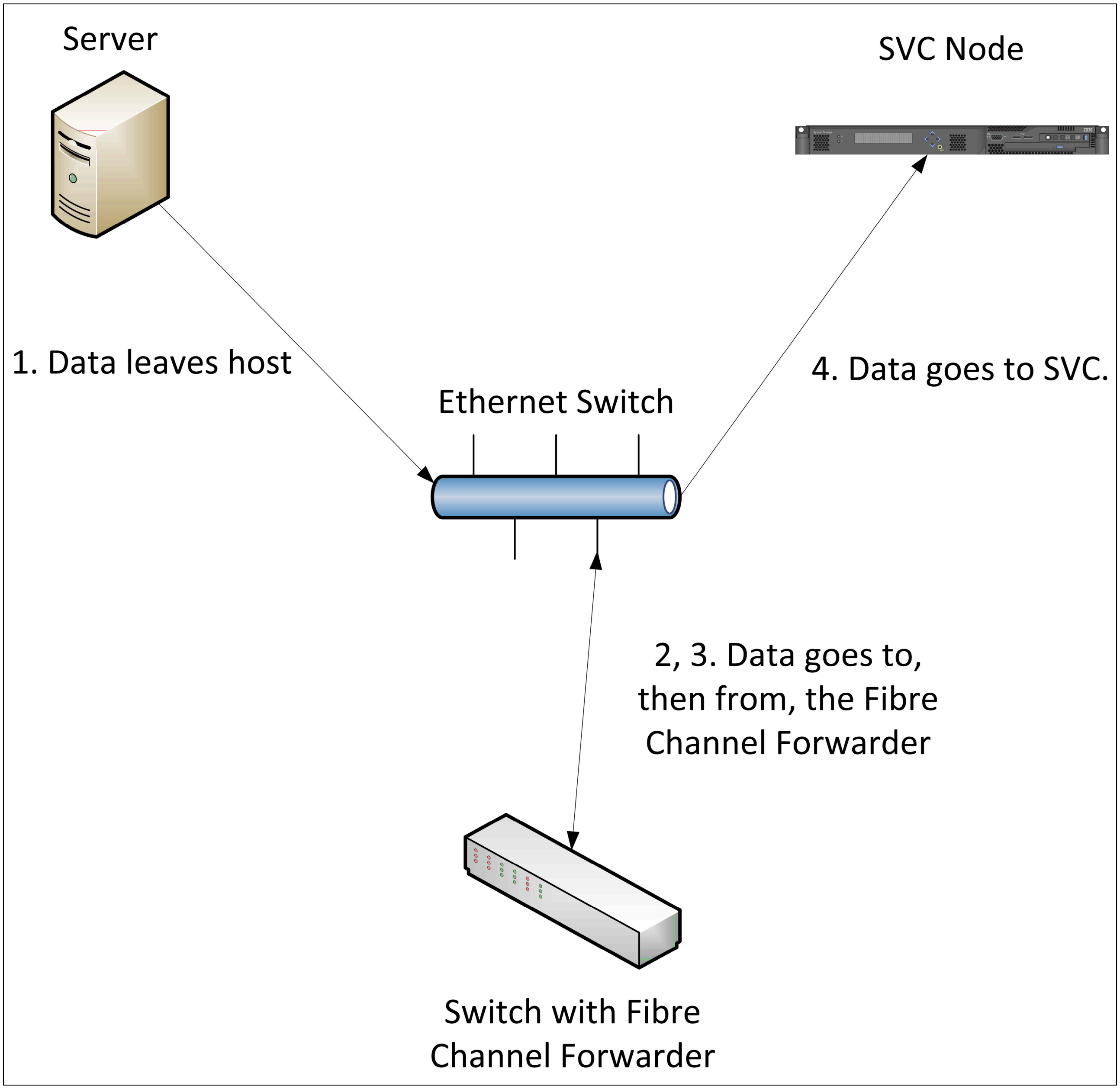


Figure 12-3 FCoE traffic flow

What impact does this have on replication? The requirement to flow all traffic through an FCF means that you must have an FCF on each end of your link if the distance between the two sites is more than approximately 5 km, unless the FCoE links are configured to be mirror-only ports.

12.3 Topology and zoning for Stretched Cluster

Although the SVC has had Stretched Cluster capability for some time, SVC V6.3 opened up new supported options for deploying a Stretched Cluster. The Stretched Cluster capability has several unique SAN topology requirements that are somewhat different from a regular SVC deployment.

For a more complete description of Stretched Cluster, refer to Chapter 7, “Volume Mirroring” on page 275.

12.3.1 Stretched Cluster and the third-site quorum requirement

For Stretched Cluster to provide resiliency advantages, the SVC must have a quorum disk that is located in a different failure domain from the two sets of SVC nodes. If the quorum disk is located such that it experiences a failure, like a circuit breaker opening, simultaneously with one of the Stretched Cluster nodes, successful failover might not occur. Putting the quorum in a different failure domain is referred to as the “extended quorum.”

You must configure the SVC to ensure that a third-site MDisk is one of the quorum candidates; the default configuration uses a portion of the first three connected storage controllers. You must also disable the dynamic quorum selection. To configure this, use the command `svctask chquorum -MDisk <mdisk_id or name> -override yes`.

The controller used for this function must be listed as supporting Extended Quorum in the SVC Supported Hardware List appropriate for your version of the SVC software. For SVC 6.4, the supported hardware list is available at the following site:
http://www-01.ibm.com/support/docview.wss?uid=ssg1S1004111

If your third-site quorum is an active/passive storage controller such as an IBM DS3000/4000/5000-series unit, each controller must be connected to both SANs.

FCIP may be used to provide the link to the quorum disk; the requirements are no different than that for Metro and Global Mirror (80 ms round-trip delay maximum). However, if FCIP is used, you must also use SAN Routing to ensure that the remote Fabric is not merged with the primary SVC fabric.

Any piece of long distance equipment used to connect to the quorum disk, for example, a DWDM, should *not* also be used to connect the split SVC nodes to each other. Otherwise, the resiliency advantages are lost.

12.3.2 “Classic” Stretched Cluster topology - no inter-node ISLs

The most basic Stretched Cluster topology involves connecting the SVC nodes directly to the fabrics at the two production sites. This solution requires that there be no ISLs between the nodes. In practice, this topology will usually involve four fabrics so that attached hosts do not need to utilize long distance links.

Only the SVC nodes themselves require the use of the direct long distance connections. These long distance connections can only be made with either dark fiber or CWDMs. DWDMs are not currently supported in this configuration.

The maximum length of the node links is dependent on the speed of the links. At 8 Gb, and using CF8/CG8 nodes, you are limited to 10 km. At 4 Gb you are limited to 20 km. At 2 Gb you are limited to 40 km. With the older 4 Gb nodes, the limits are 4 km, 8 km, and 16 km at 4 Gb, 2 Gb, and 1 Gb, respectively. These limits are imposed by the available FC buffer credits within the SVC nodes. Also keep in mind that the particular distance extension technology you choose might impose further limitations.

FCIP, or other similar IP-based distance extension technologies, cannot be used in this configuration for the inter-node connections. However, they may be used for the Quorum. Hosts and the Quorum disk may each use up to one ISL hop (or FCIP tunnel) to access the SVC.

With this configuration, two ports on each SVC node in the split group attach to the local fabrics and two ports attach directly to the remote fabrics. An example of this topology is shown in Figure 12-4.

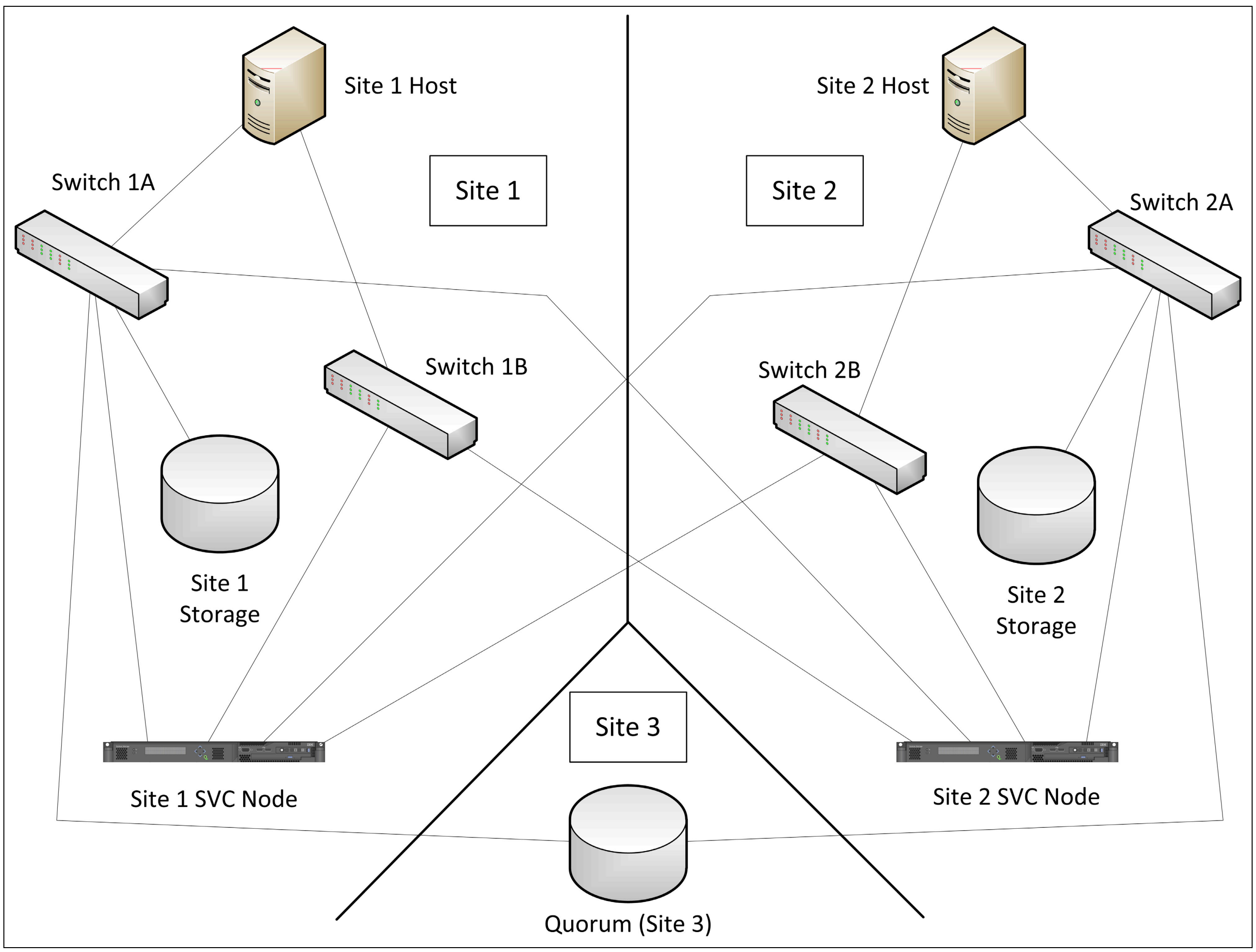


Figure 12-4 Classic Stretched Cluster topology (no ISLs)

Switch Buffer adjustments - Brocade Fabrics only

Without configuration changes, Fibre Channel links that travel more than 2 km (at 8 Gb) might start to suffer from throughput limitations due to the default Fibre Channel flow control settings on Brocade switches.

For 8 Gb SVC nodes connected to SVC fabrics, you must increase the amount of buffer resources available to the SVC in switch ports attached to the long distance links.

In ISLs, this is done using the **`portCfgLongDistance`** command, but that option only works on ISLs. For this Stretched Cluster solution, you must manually adjust the number of buffers in the switch using the **`portCfgPortBuffers`** command. The syntax for this command is: **`portcfgPortBuffers --enable <port number> 45`**. The 45 refers to the number of buffers we suggest for 8 Gb SVC HBAs.

12.3.3 Stretched Cluster with inter-node ISLs

Starting with SVC version 6.3, instead of connecting the nodes directly over the long distance link, the links between the Stretched Cluster nodes can be implemented over ISLs. Figure 12-5 illustrates an example.

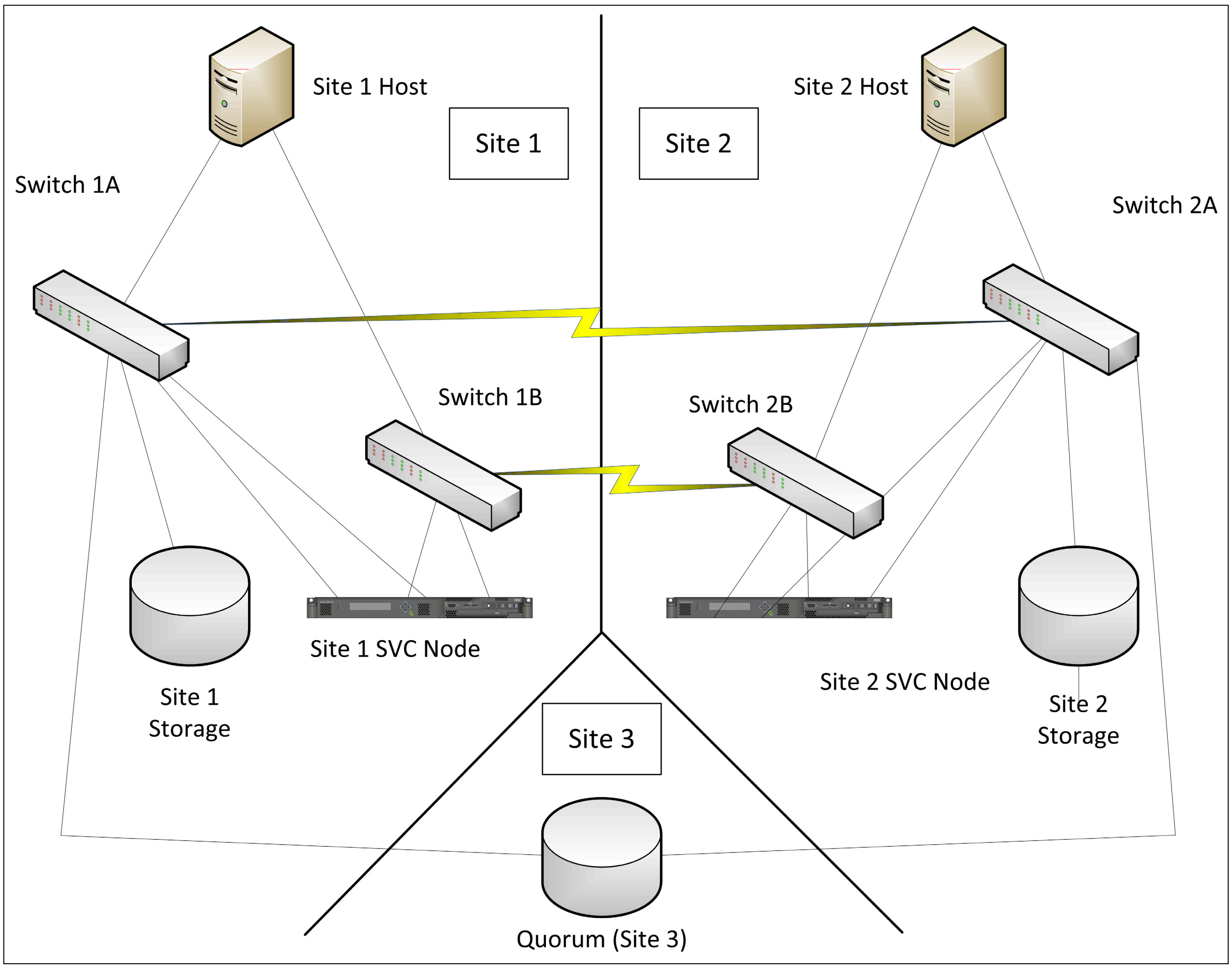


Figure 12-5 Stretched Cluster with ISLs

Be aware that there are still significant limitations to this configuration. The most significant limitation is that the inter-node link must be on a separate SAN from your other traffic. You may implement this with either separate equipment, or by using Cisco VSANs or Brocade Virtual Fabrics.

Only a single ISL may separate the two sets of fabrics; you cannot configure a multi-hop SAN to bridge the two sites. The ISL is permitted to use any type of long distance FC (or FCoE) solution supported by your switches; this may include Dark Fiber, CWDM or DWDM. FCIP (or any other IP-based solution) is still not permitted for the inter-node link.

You must have one full-bandwidth ISL, not shared by any other device, per I/O group on this private fabric. You must use the link bundling technology made available by your switch vendor: for Brocade Fabrics, use Trunking, and for Cisco Fabrics, use Port Channels. The link to the third-site quorum might be either an ISL, or it might use IP-based distance extension technology.

When used for disaster recovery, the recommended length limits for the inter-node link are the same as the recommendations for Metro Mirror, up to 300 km. When used for data mobility solutions, the maximum recommended distance is approximately 150 km due to the increased number of round-trips necessary during write operations versus what is needed for Metro Mirror. This is a SCSI protocol limitation, not an SVC limit.

12.4 WAN connectivity for replication

WAN must be planned and monitored carefully to ensure a successful replication solution.

12.4.1 WAN connectivity types

There are two main, but differing, ways to connect two sites to each other: Dense Wave Division Multiplexing (DWDM) and Complex Wave Division Multiplexing (CWDM), as explained here.

Dark fiber: DWDM and CWDM

The simplest way to connect two sites to each other is by using a long length of fiber, or products that will let you mutliplex multiple signals onto a single large strand. The two multiplexing technologies in use are Dense Wave Division Multiplexing and Complex Wave Division Multiplexing.

- DWDMs function by taking incoming data streams, sorting them among various output channels, and modulating the output channels into discrete wavelengths that are then transmitted over a single fiber.
- CWDMs rely on a set of special transceivers, placed within your SVC and switch, that each generate a different wavelength. The CWDM itself is simply a passive prism that combines the various wavelengths onto a single fiber.

These technologies offer the benefit of generally delivering large amounts of low latency bandwidth in a consistent fashion. However, the cost of purchasing such a link and the equipment to utilize it might be prohibitive to some enterprises.

Extended Distance Fabric settings (Brocade SANs only)

In addition to length limitations imposed by the cabling, Brocade users must also be concerned with limitations imposed on ISLs by the default port configurations. (This section applies to ISLs only; for distance adjustments in Stretched Cluster solutions without inter-node ISLs, refer to 12.3.2, "“Classic” Stretched Cluster topology - no inter-node ISLs" on page 479.)

In Fibre Channel, flow control is accomplished through a mechanism known as “buffer credits.” The number of buffer credits available to a link puts a limit on a link’s maximum length. This maximum length varies by port speed. The presence of a CWDM or DWDM does not increase or decrease this allowed link length. For more detailed information about buffer credits, see “Fibre Channel buffer credit flow control” on page 496.

At the default settings, the maximum usable length of a Brocade link is 5 km at 2 Gbps, up to 2 km at 4 Gbps, and up to 1 km at 8, 10, and 16 Gbps. If your link is longer than these distances, you need to utilize **portCfgLongDistance** to configure the appropriate link length. At the time of writing, if the length is longer than 10 km, you need to purchase and install an Extended Fabrics license. There are also limits on the number of extended-length links you can configure per switch port group; the exact number depends on the length of the links and the particular model of port card or switch.

For further information, refer to the *Brocade Fabric OS Administrator’s Guide* appropriate to the level of Brocade software you are running.

FCIP

Both Brocade and Cisco have products that will encapsulate FC traffic into a TCP/IP session and send it out over a Gigabit Ethernet link into a standard IP network.

Because most organizations already have an IP link between sites, this form of connectivity is often much less expensive than dark fiber or a CWDM or DWDM. However, if the link is not managed properly, it can deliver inconsistent amounts of bandwidth or suffer from unpredictable latency. A poorly managed FCIP link might cause an unstable mirroring environment and might even have a performance impact on your hosts.

The network “cloud” providing the IP links between the two sites should generally be part of a dedicated leased line. Attempting to run an FCIP link through a VPN tunnel within an Internet link will likely result in unstable amounts of latency and bandwidth.

Shared IP links

Most FCIP installations will run the FCIP connection through a shared intersite link. It is important to configure the network between the FCIP routers to ensure that the FCIP link *always* has access to the bandwidth you have configured the FCIP tunnel for. In a Brocade environment, this is the “committed rate.” If you are using Adaptive Rate Limiting, it is the “minimum committed rate.” In a Cisco environment, it is the “minimum bandwidth.”

12.4.2 WAN redundancy

As with any IT solution meant to achieve proper redundancy, you need to ensure that if you lose one WAN link, the remaining links are capable of carrying the full load of your mirroring. If losing one link means you do not have sufficient bandwidth to handle your complete production load, you have not achieved redundancy at all. Instead, you might end up with a significant host performance impact in the event of an issue with the WAN link. This is especially applicable to Metro Mirror installations, where any increase in latency will be keenly felt by your applications if you have not factored in full redundancy.

12.4.3 Link quality

This section discusses latency and packet/frame loss considerations.

Latency

The maximum allowed link round-trip latency between the Master and Auxiliary clusters, or between the nodes and the third-site quorum disk in a Stretched Cluster solution, is 80 ms. That being said, the synchronous nature of Metro Mirror and Stretched Cluster Volume Mirroring places practical limits on the maximum amount of latency that can occur before there is an unacceptable performance impact to your end applications. Most applications will not tolerate anywhere near 80 ms of I/O delay.

Long distance links that are composed of dark fiber or CWDMs or DWDMs can generally be expected to deliver round-trip latency of approximately 1 ms per 100 km.

> **Link distance consideration:** The 1 ms per 100 km latency mentioned here refers to the *actual* link distance, which might be much greater than the geographical straight-line distance between the two sites. To determine the actual link distance, contact the provider of the link.

FCIP links will generally have round-trip latency significantly higher than a link of equivalent distance over plain fiber due to IP routing delays. Certainly it will not be less than 1 ms per 100 km, but there is no predefined upper bound on how much latency an IP network might introduce. It is useful to have the network team perform pre-deployment latency tests under maximum load over the span of at least one week to ensure that latency and packet loss measurements remain within acceptable limits at all times.

Packet/frame loss

The SCSI protocols used for storage traffic on a SAN are built around the idea of essentially zero frame loss and in-order low latency delivery. Although error recovery due to packet loss or out-of-order delivery is possible, the methods for doing so introduce a much higher performance impact than most other TCP/IP-based applications. As a result, packet loss results in severe performance degradation well out of proportion to the number of packets actually lost. A link that is considered “high quality” for most TCP/IP applications might be completely unsuitable for remote mirroring.

In FCIP environments, field experience has shown that a link with packet loss of 0.02 percent (two one-hundredths of one percent) is marginal, and there will likely be significant constraints on maximum usable throughput if packet loss reaches approximately 0.2 percent. In Metro Mirror or Split I/O Group environments, 0.2 percent loss is also likely to cause host latency to rise to unacceptable levels. Global Mirror will usually go offline in such circumstances.

In dark fiber or CWDM or DWDM environments, the most common issue affecting Remote Copy installations is frame corruption. If, in the switch, you see evidence of more than 50 or so CRC errors on any given link in a day, it is advisable to perform appropriate physical layer troubleshooting on the link to bring the number of CRCs per day down to less than 10. The Fibre Channel specifications allow for much higher error rates than this, but a link that is logging more than a handful of CRC errors a day is likely to get worse over time, and possibly suffer a sudden complete failure.

For more information about finding and diagnosing issues related to IP packet loss, refer to Chapter 13, “Troubleshooting Replication Family Services” on page 491.

12.4.4 Bandwidth planning

Determining the amount of WAN bandwidth necessary is a difficult task, and the bandwidth needed differs depending on which mirroring technology you select. Although it is possible to make a guess at it, this often leads to too much or too little bandwidth.

Because of the way in which Fibre Channel flow control operates, insufficient WAN bandwidth might have severe affects on the performance of the SVC and all attached hosts (especially in Metro Mirror environments), so it is better to err on the side of having too much bandwidth versus not enough bandwidth.

> **Unit of bandwidth measurement:** Most storage monitoring tools and storage products (as well as this book) use Mega*bytes* per second (MBps) as the standard unit of bandwidth measurement. Most network monitoring tools (and most network engineers) think of bandwidth in Mega*bits* per second (mbps.)
>
> When communicating between teams, it is important to notice the difference, or your bandwidth provisioning might be off by as much as an order of magnitude.

Initial traffic measurement

To do the best job of predicting needed bandwidth, use a storage monitoring tool such as IBM Tivoli Storage Productivity Center for Disk. Using sampling intervals no longer than five minutes, use a spreadsheet to sum together the write throughput of every volume that you want to mirror, and use as a starting point the largest of those sums. Make sure you take measurements over an entire business cycle to ensure you capture the true peaks for your installation. For instance, if a twice-monthly payroll job accounts for a large amount of load, make sure that your bandwidth data includes that job.

To this value, a general suggestion is to add a “cushion” of approximately 20 percent to account for errors that might be introduced by measurement sampling intervals, overhead, a small amount of growth, and unexpected busts of write activity. You might want a larger cushion for Metro Mirror installations, or a smaller one for Global Mirror. Note that this is only a starting point; based on the details of your particular workload, you might need more or less. Generally, Metro Mirror users will want a larger cushion because there is a higher likelihood that there will be usage spikes that will be masked by caching in a Global Mirror.

Post-deployment, we strongly suggest that the network team continue monitoring the WAN bandwidth in use, and purchase more bandwidth as the link nears capacity. Keep in mind that although background copy is not a constant activity, you might need to perform a sync operation at any time. Therefore make sure that your links have sufficient bandwidth to run a background copy job in addition to your peak workloads. For instance, if your configured background copy bandwidth is 50 MBps, and your link has a capacity of 80 MBps, strongly consider the purchase of additional bandwidth when your peak write workloads begin to exceed approximately 25 MBps (50 MBps background bandwidth + 25 MBps write I/O + 5 MBps cushion = 80 MBps).

Peak versus average performance rates

When deciding how much bandwidth to procure, you must use *peak* bandwidth numbers, as opposed to an average. If at any point during your mirroring deployment, even for just a few minutes, bandwidth exceeds what you have procured, you might encounter performance or stability issues. This is why it is important to examine performance data over an entire business cycle when choosing bandwidth amounts. For example, if you have a large monthly report on the third Thursday of every month, make sure that your performance data covers that period.

For Metro Mirror, use a sampling interval as short as possible so as to not miss any spikes. And even then, you might miss a sudden spike; for this reason you might want a cushion larger than 20 percent with Metro Mirror installations if you have performance-sensitive applications.

Background copy bandwidth

The amount of bandwidth to allocate to mirror synchronization traffic solely depends on the amount of data you are mirroring, and your business requirements as to how long it should take to completely copy that data from one site to another. This bandwidth is specified when you create a partnership between two clusters, and it is shared among all mirrored volumes.

Bandwidth parameter consideration: The bandwidth specified when you create a cluster partnership refers only to the *background* copy bandwidth. Your "foreground" I/O (your actual write workload from your applications) is in addition to the bandwidth specified during partnership creation.

For instance, if you have determined your peak write traffic is 50 MBps, but to meet your business requirements you need to synchronize the two sites at 20 MBps, then you will set the "partnership bandwidth" parameter to 20 MBps and make sure you have 70 MBps of bandwidth available over your WAN.

To emphasize, the bandwidth parameter in the cluster partnership configuration places *no limit* on how much bandwidth the SVC will attempt to use to mirror your application I/O.

Bandwidth and Global Mirror with Change Volumes

If you are using Global Mirror with Change Volumes, you do not need to be as concerned about a temporary spike in I/O activity causing performance issues in your hosts due to congestion.

However, if the amount of I/O over one Global Mirror Cycle exceeds the bandwidth you have to transport it, you will skip one of your cycles while the cluster finishes transmitting the data from the previous cycle. This will cause the remote copy to be older than what your configuration was set up for.

Data compression

Many SAN extension products utilizing TCP/IP offer data compression to lessen the amount of needed WAN bandwidth. Although you can certainly take this into account when planning your mirroring installation, try to determine how compressible your data is before depending on it to reduce the amount of bandwidth you must purchase.

For instance, your average database is quite compressible, and compression ratios of at least 2:1 are quite common. Conversely, data such as images or video are often not compressible and you will obtain no benefit from WAN compression.

For an idea of the usable compression that can be obtained with your data set, you might be able to perform various tests ahead of time. For a file server, simply compress a representative sample of files with any operating system file compression utility. Although something that compresses an entire file will do a better job than a network streaming device, it should still give you a general idea of what is possible. For a database server that uses direct disk access, you might be able to get advice from your application vendor.

After your FCIP tunnel is in operation, the routers themselves will be able to inform you of the Fibre Channel data totals versus the totals transmitted over your FCIP link. If using these statistics, make sure your measurements are done after synchronization is complete. During synchronization, the SVC might be transmitting large amounts of empty blocks; these are highly compressible and might artificially increase the reported compression ratios during this time.

Adaptive Rate Limiting (Brocade) or TCP Window Management (Cisco)

Both Brocade and Cisco environments have features that will allow your FCIP connection to adapt the bandwidth in use to the current available bandwidth on the link. They both involve setting a minimum guaranteed value, and then setting the maximum to the largest possible WAN bandwidth.

This might be useful to provide a bandwidth cushion for increases in workload. However, because any bandwidth beyond the minimum is not guaranteed, allocate more bandwidth to FCIP, and adjust minimums accordingly, if your network monitoring tools show that you are approaching or passing your current guaranteed bandwidth, and adjust the SAN Router minimums accordingly. Do *not* rely for the long term on any non-guaranteed bandwidth. Otherwise, your mirroring solution might become unstable as it grows.

WAN Quality of Service configuration

Most installations of FCIP links will utilize WAN links that are shared with other applications. If the WAN link is shared, it is important that the network routers be configured to ensure that the necessary FCIP bandwidth is available at all times. Although you may configure the WAN so other network services use bandwidth not currently needed by the FCIP tunnel, the FCIP link should have priority bandwidth allocation, up to the maximum you have configured in the FCIP tunnel.

WAN link monitoring

A vital part of any mirroring solution deployed over a WAN is constant monitoring. On dark fiber, CWDM, and DWDM solutions, use SAN monitoring tools to ensure that the number of CRC errors in a day is below 10. Also look for Loss of Signal and Loss of Sync errors.

With FCIP solutions, use network monitoring tools (as opposed to SAN tools) to monitor the bandwidth utilization of the link. Procure more bandwidth if the utilization rises above approximately 80 percent of your configured bandwidth guarantees at any time.

For any utilization monitoring, use no greater than a 5-minute sampling interval. A larger sampling interval will mask problems.

Inter-cluster heartbeat

The SVC Configuration Guide makes reference to the bandwidth needed for the inter-cluster heartbeat. The most bandwidth this heartbeat can possibly use is 1.2 MBps. However, when actual storage traffic is transiting the link, the heartbeat is not used at all. Because 1.2 MBps is not enough for any practical mirroring solution, the inter-cluster heartbeat does not need to be considered in bandwidth calculations.

12.5 SAN routing with mirroring

When connecting two different sites to each other, it can enhance fabric resilience if you prevent instability on the WAN link from causing performance or stability problems with the local traffic on each end. Either Brocade SAN Routing (either internal or through an external unit) or Cisco Inter-VSAN Routing (IVR) can achieve this goal.

Routing consideration: The SAN routing discussed in this section is distinct from FCIP WAN links. Although one is often used along with the other, this is not a requirement. SAN routing may be used with or without FCIP, and FCIP may be used with or without SAN routing.

12.5.1 SAN routing with Brocade products

For information about configuring FC-FC Routing in a Brocade fabric, refer to the Redbooks publication *IBM System Storage b-type Multiprotocol Routing: An Introduction and Implementation*, SG24-7544.

Figure 12-6 illustrates an appropriate isolated SAN topology with external SAN routers.

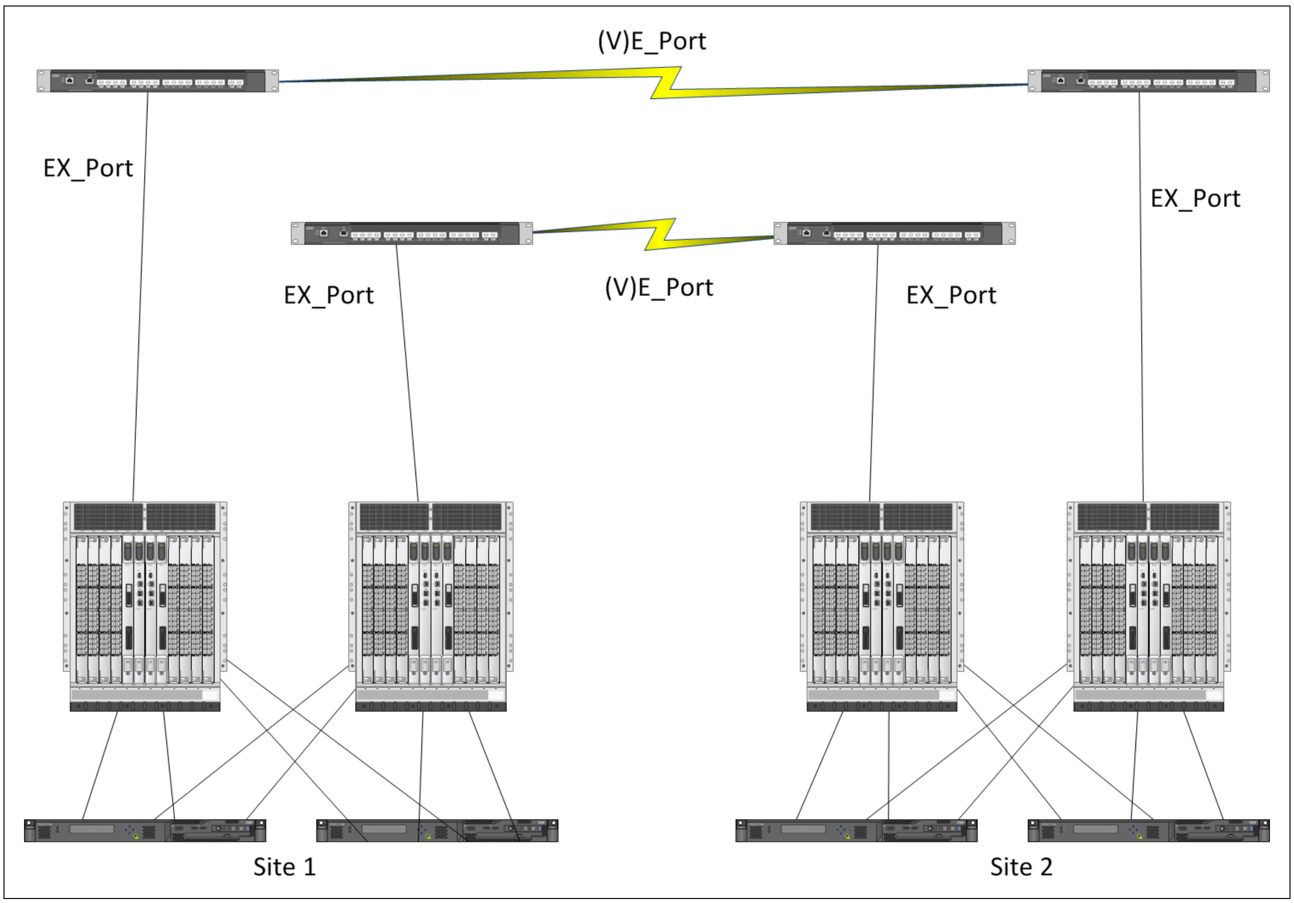


Figure 12-6 Brocade routing design with routers

If you are using the internal FC-FC routing service only, and not external routers, your topology will look as shown in Figure 12-7 on page 489.

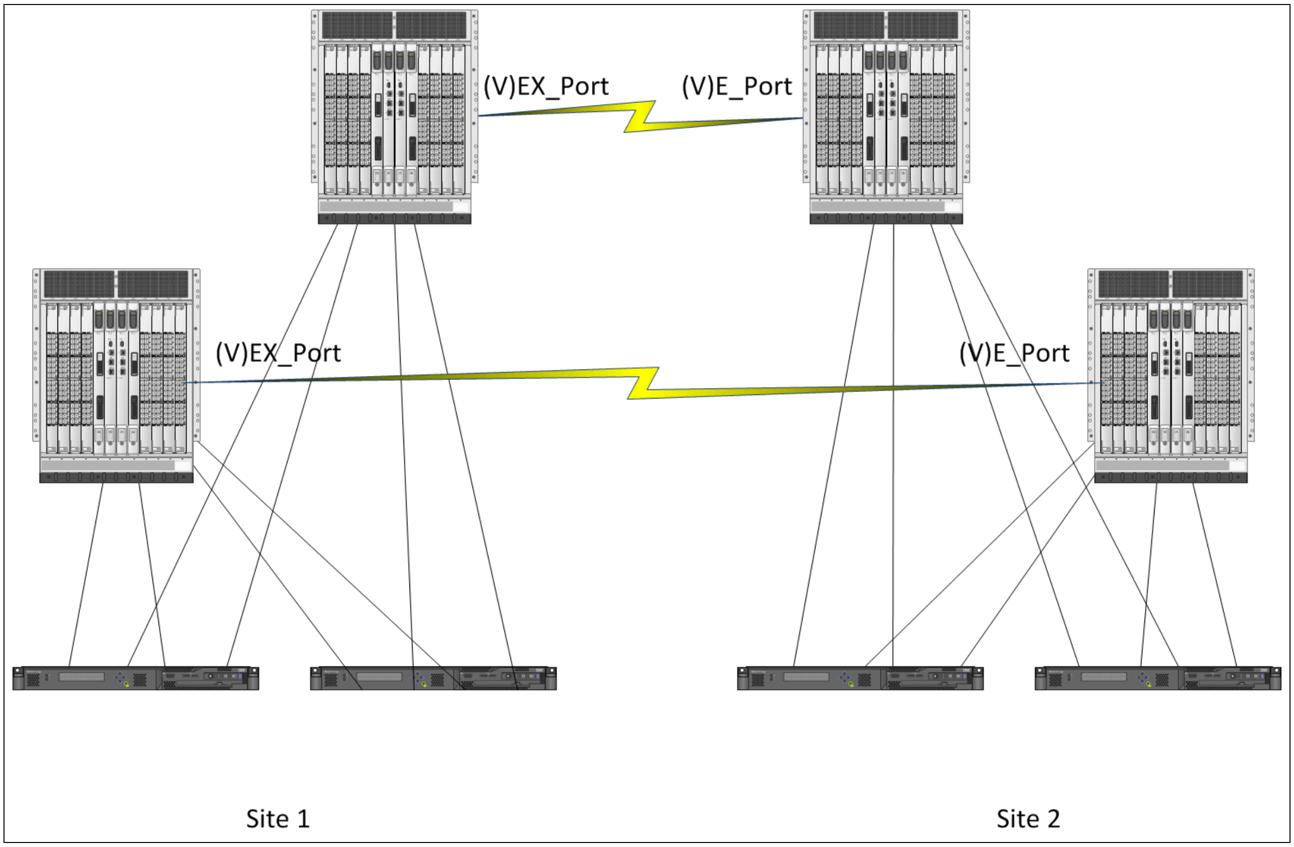


Figure 12-7 Brocade FC-FC internal routing service

12.5.2 SAN routing with Cisco Inter-VSAN Routes

For instructions about configuring Cisco IVR, refer to the Redbooks publication *IBM/Cisco Multiprotocol Routing: An Introduction and Implementation*, SG24-7543. Figure 12-8 on page 490 illustrates the configuration to aim for.

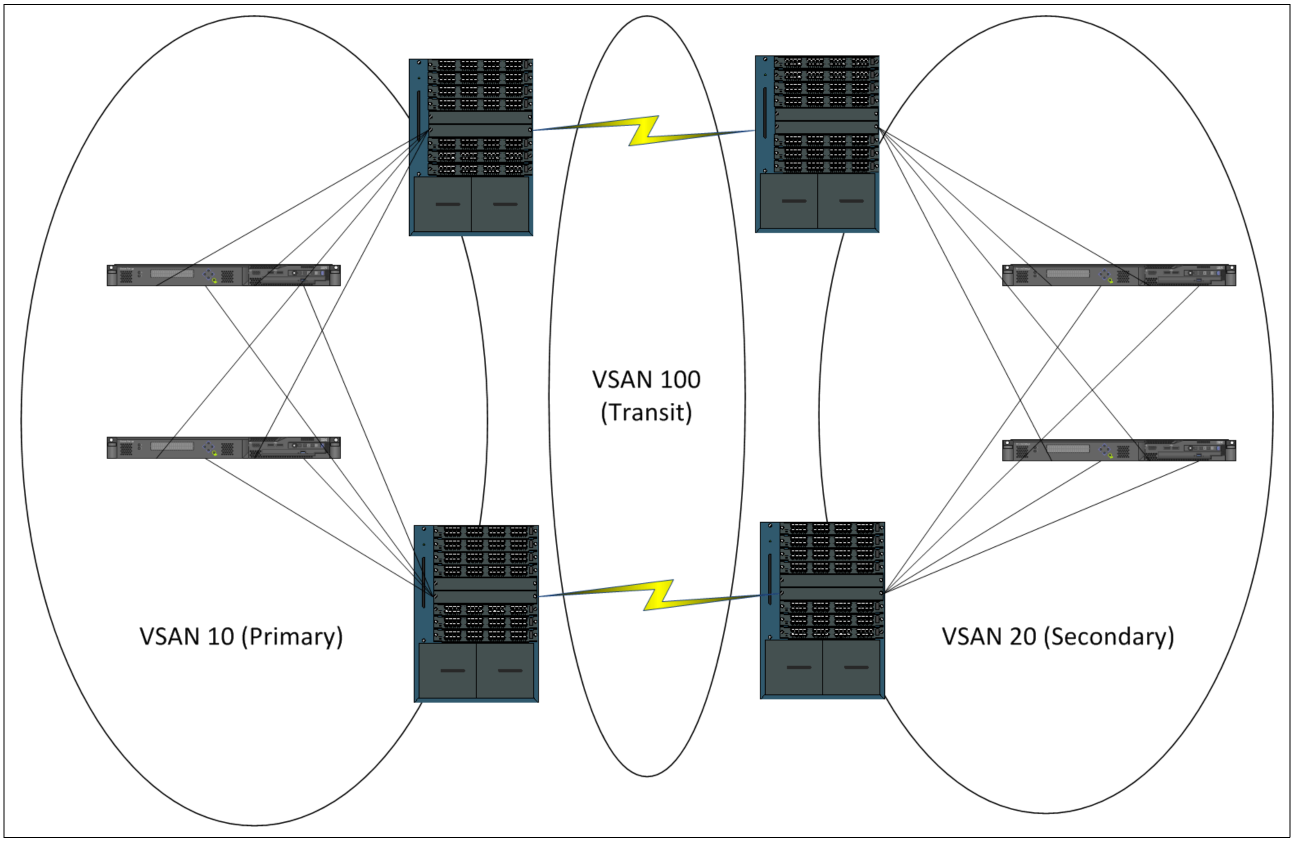


Figure 12-8 Example Cisco IVR mirror topology

13

Troubleshooting Replication Family Services

Diagnosing and correcting issues with your replication services as quickly as possible saves you time, money, and a support call.

The following troubleshooting topics are discussed here:

- Basic replication diagnostics, including explanations of common replication error codes and methods to help you collect SVC performance data
- Basic replication performance troubleshooting tips
- SAN diagnostics for replication

13.1 Basic replication diagnostics

The following sections explain the meaning of various SVC error codes and also describe methods for collecting information about the most basic part of Replication Family troubleshooting, which is performance data.

13.1.1 Common replication error codes

This section identifies common replication error codes and explains the steps needed to troubleshoot them.

1710

There are too many Cluster Partnerships. The number of cluster partnerships has been reduced.

This message will be posted if your entire clustering solution has more than four partners, either directly or indirectly. For instance, if Cluster A is partnered with B, and B is partnered with C, and C is partnered with D and E, then A's cluster partnership count is four. (A, B, C, D, and E.) You must re-architect your solution to bring down the number of direct or indirect partnerships.

1720

In a Metro Mirror or Global Mirror operation, the relationship has stopped and lost synchronization, for a reason other than a persistent I/O error.

The most common cause of this error is an intercluster SAN/WAN link that has gone down. Examine your SAN switch and router logs for evidence of link trouble.

For Brocade solutions, examine the output of the command **`fabriclog`** (or, in older versions, **`fabstateshow)`**. In Cisco solutions, examine **`show logging log`**.

Other common causes of this error include:

- Destination cluster stability

  Examine the event logs of the destination cluster to determine if it had problems around the time of your error condition.

- Metro Mirror Destination Cluster Performance

  Check the performance of the destination cluster. Ensure that the remote mDisks are responding well. What qualifies as "responding well" will vary depending on precisely what you are doing with the SVC. The best technique is to compare it with earlier performance statistics, if available.

  Also check that the CPUs of the remote nodes are not running over 80 percent.

1910

A FlashCopy mapping task was stopped because of the error that is indicated in the sense data.

The most common cause of this error is stopping a related FlashCopy if you are using cascaded FlashCopy. For more information about Cascaded FlashCopy, refer to 5.2.5, "Cascaded FlashCopy" on page 157.

1920
Global and Metro Mirror persistent error.

This error message is the most common performance-related mirroring error. Many different conditions might cause this error. They include:

- Insufficient WAN bandwidth. If this is the problem, it can come from one of two sources:
  - A poor quality or overconfigured WAN link. To check this, refer to 13.4, “FCIP congestion troubleshooting” on page 500.
  - Driving more data than your WAN link can handle. If the “backpressure” from the WAN persists long enough or harms your I/O performance, the SVC/V7000 will terminate the mirror as a protective measure.
- Performance issues at the remote cluster. This is usually caused by poor destination controller response time, but can also be triggered by CPU overload in the remote cluster. Check the remote cluster for disk performance issues or unfixed errors. Also examine the logs of the destination storage controller.
- The destination volume is a FlashCopy source in the Prepared state. Volumes in the Prepared state have write cache disabled, often significantly reducing their write performance.
- The primary volume or storage controller might be overloaded.

If you encounter this error, examine your performance data for an unusual workload increase just prior to the error (workload in this case referring to either I/Ops or MBps).

13.1.2 Collecting SVC performance data

Most issues with replication will require a performance monitoring tool that can produce graphs of SVC performance statistics. Much of this chapter will assume you have access to these statistics and a tool for turning them into graphs.

The IBM tool that is supported for this use is TPC for Disk, and it is highly advisable to use it. However, if you do not have TPC for Disk, the SVC does produce raw performance statistics in XML format and there are unsupported free utilities available from IBM that may be used to parse them into a more useful form:

- svcmon

  This utility is a full-featured performance parsing utility. However, it requires an installation of postgres, perl, and other utilities, depending on whether you install it on Windows or Linux. Information about this utility is available at:
  `https://www.ibm.com/developerworks/mydeveloperworks/blogs/svcmon/entry/introduction?lang=en`
- qperf

  This utility is useful for performing basic SVC performance checks. All it requires is an OpenSSH client and a korn shell (either a Linux system or a Windows system with cygwin). More information about this utility is available at:
  `http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/TD105947`

13.2 Basic replication performance troubleshooting tips

The rules for quality replication performance are similar to those for any SVC/V7000 solution. For general SVC/V7000 performance tips, refer to *SAN Volume Controller: Best Practices and Performance Guidelines*, SG24-7521.

For replication-specific performance tips, however, see the following sections.

13.2.1 First steps

If you encounter a performance issue that you perceive is related to replication, shut down the replication, if possible, and see whether the problem disappears (this might not be possible for many solutions). If the problem does not go away, then the performance issue is caused by something other than replication itself.

If the problem does disappear, examine the SVC and SAN performance statistics to attempt to discover the bottleneck that is affecting your performance.

13.2.2 Workload analysis

As with any performance issue, your primary task is to determine "what changed?" Did your replication workload increase? Did your WAN quality drop? Did you introduce new workload on shared storage controllers? Did the application administrators start a new periodic job? Did you recently install a new management utility? It is rare for performance to "just drop," so being able to answer these questions can help you resolve a performance problem.

When performing your analysis, keep in mind the relationship between workload and response time. Basically, when a system is running with its limits, response time slowly increases with workload. When a system is run faster than the limits, response time increases rapidly with any additional load. This means that small changes in workload or capacity can result in surprisingly large decreases in performance if the system was previously running near maximum capacity.

13.2.3 Source and destination storage controllers

With all types of replication, generally the destination disk *must* be able to handle your particular write performance requirements (both MBps and I/Ops) as well as the source disk does. If this is not the case, the performance of the source disk will be restricted to the write performance of the destination disk. You do not have to have identical storage controllers at the destination site (although this is certainly the easiest way of assuring adequate performance), but whatever solution you do develop needs to match the write performance of the source.

If your solution will be used for disaster recovery then the secondary controller should also be able to match the read performance of the primary, unless you can accept a performance decrease during a failover and recovery scenario, or a failover or recovery scenario.

13.2.4 SVC/Storwize V7000 CPU utilization

If the SVC/Storwize V7000 CPU is overloaded for whatever reason, it can have unpredictable performance consequences. A check of the CPU utilization statistics is an important part of most performance troubleshooting. If using Metro or Global mirror, remember to check the

CPU utilization at the secondary site. All replication functions use some CPU versus replication not being enabled.

13.2.5 WAN performance (Metro and Non-Cycling Mode Global Mirror)

For Global Mirror and Metro Mirror solutions, adequate WAN bandwidth is important. Write I/O workloads exceeding the available WAN bandwidth might cause performance impacts to the applications running on your primary. Metro Mirror, due to its synchronous nature, is more susceptible to this issue.

“Bursty” workloads (those susceptible to short bursts of intense write activity) can place special burdens on WAN links. Such burdens can be difficult to perceive ahead of time with performance monitoring tools. If you know you have an application that is subject to sudden loads, make sure you have adequate resources to handle such loads.

For Metro Mirror, you will need a larger cushion. For Global Mirror, you will need a smaller cushion, although Global Mirror will experience an outage if the bandwidth shortage becomes large enough or persists long enough. Cycling Mode Global Mirror will not suffer a performance penalty due to inadequate WAN bandwidth, but you might “miss” a cycle, thereby causing you to fail to achieve your Recovery Point Objective.

As with all storage solutions, the more information you have about your applications, the better estimates you will be able make as to their performance requirements. Misunderstanding the workload over time can lead to overprovisioning or, more commonly, underprovisioning your resources.

13.3 SAN diagnostics for replication

Proper operation of mirroring depends on a well-functioning SAN. If you are experiencing poor host performance whenever mirroring is enabled, or are seeing 1720 or 1920 errors, examining the link between the two clusters is a good starting point for your troubleshooting efforts.

13.3.1 Fibre Channel flow control overview

To understand SAN diagnostics as they relate to mirroring, you must first understand how the Fibre Channel protocol stack performs flow control. There are two forms of flow control in Fibre Channel: SCSI-FCP and buffer credit, as explained in the following sections.

SCSI-FCP flow control

To prevent internal buffers from being overrun, SCSI has a mechanism that will prevent the device on the other end from transmitting too much data at once. For read commands, this is done by simply not requesting more data than there are allocated buffers in the SCSI initiator. For write commands, the SCSI Target must send a frame called a XFer_Rdy (Transfer

Ready), giving the initiator permission to transmit the data. Figure 13-1 on page 496 illustrates this packet flow.

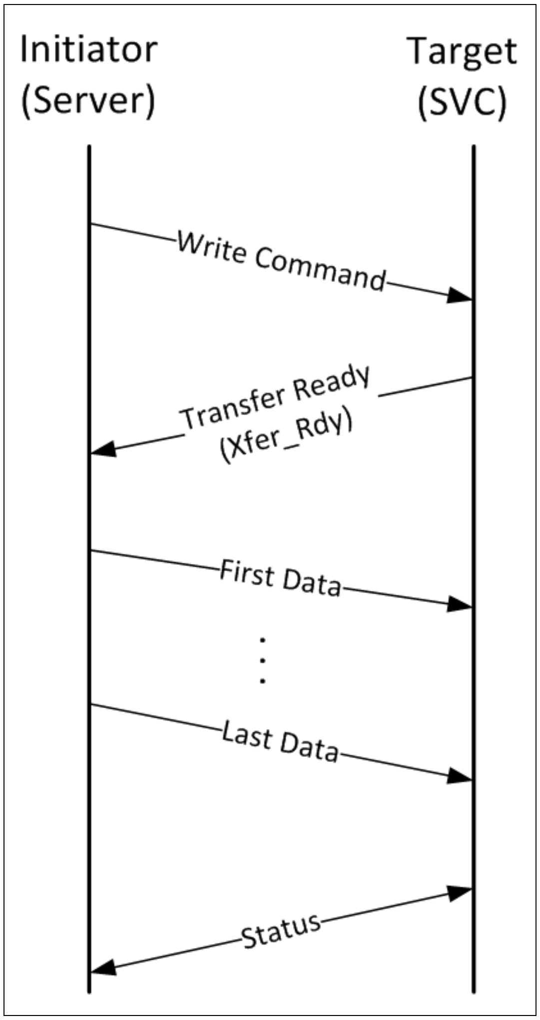


Figure 13-1 SCSI-FCP packet flow

This method of flow control is used with both regular Fibre Channel and FCoE.

Because of the additional one-way transmission delay waiting for the XFer_Rdy, we strongly suggest that you make sure that preferred node for the primary volume in a Stretched Cluster configuration be assigned to the local node. This is also why basic SVC/Storwize V7000 configuration guidelines suggest that a storage controller never be placed at a remote site.

This flow control mechanism is only relevant to traffic to and from hosts and to and from storage controllers; the SVC/Storwize V7000 uses a proprietary flow control protocol for inter-node and inter-cluster traffic that eliminates the need for the XFer_Rdy.

Fibre Channel buffer credit flow control

To control the flow of traffic on the links themselves, Fibre Channel makes use of a mechanism called “buffer credits.” Each Fibre Channel port has a certain number of internal buffers available. When a port transmits a frame, it decrements an internal counter for tracking buffer credits. When the receiving port receives and processes the frame, it sends a piece of data called the R_Rdy (Receive Ready.) When this R_Rdy is received, the internal credit counter is incremented back up.

During times of congestion (for instance, an overloaded WAN link), the receiving port will withhold credits from the other end. When the available credit counter reaches zero (0), the transmitting port simply stops sending until it finally receives a credit from the receiving port. Figure 13-2 illustrates this traffic flow.

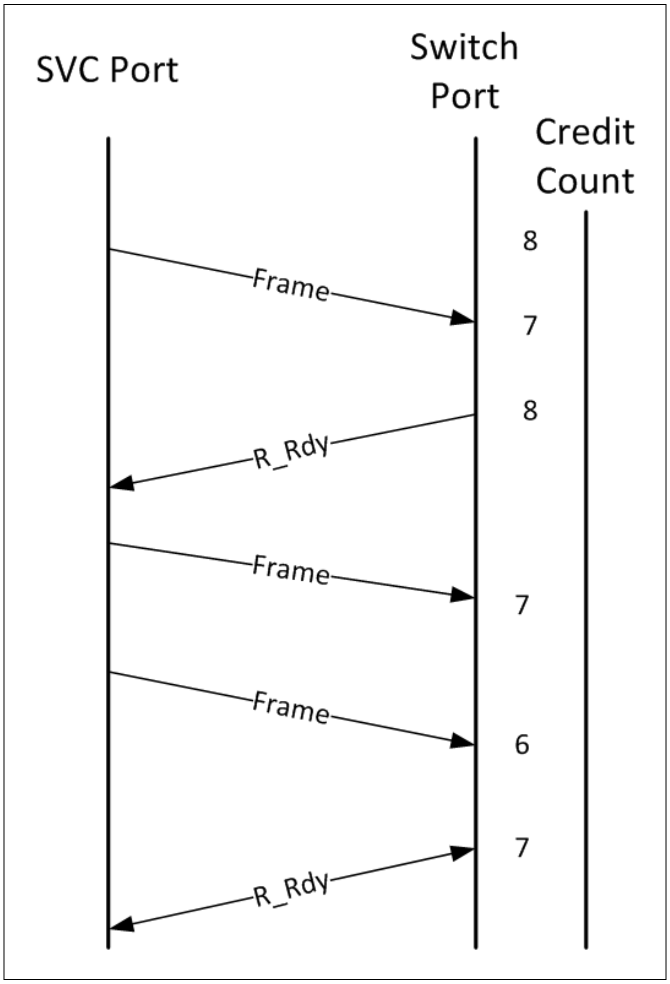


Figure 13-2 FC Frame traffic flow

Buffer credits are used strictly on a port-to-port basis; there is no end-to-end flow control used except for the mechanisms provided by SCSI-FCP and discussed in "SCSI-FCP flow control" on page 495.

SVC buffer credit information

TPC for Disk and other SVC performance tools that make full use of the SVC performance data have the ability to retrieve buffer credit starvation statistics. These statistics are presented as the number of microseconds over the sampling interval during which the SVC had no credits. This can be easily converted in a spreadsheet to a more useful percentage. For instance, if you use a 5-minute sampling interval, divide the number TPC gives you by 300 000 000 (the number of micro-seconds in 5 minutes) to obtain the ratio of time the port you are analyzing had zero credits.

SAN Credit Monitoring on Brocade Fabrics

The generation of Brocade hardware capable of 8 Gb or higher speeds have a capability called "bottleneck monitoring." If your solution has an FC ISL somewhere between the two SVC clusters, the bottleneck monitoring function can be configured to alert you when the link between the two sites is experiencing credit starvation. As with any performance statistic, some fine-tuning is necessary to alert on problems, but not constantly alert on false positives.

For more details, consult the *Fabric OS Administrator's Guide* appropriate for your code version.

Analyzing buffer credit statistics

There are several important points to keep in mind when analyzing buffer credit statistics:

- Remember that credit starvation is *not* usually an error condition. Although high levels of credit starvation over an extended period might lead to performance issues, analyze the information in the context of your entire solution that is helpful for troubleshooting and trend analysis; it should *not* used as a primary performance measurement.
- Simply because the SVC (or switch) had no credits does not necessarily mean that it had additional information beyond that point to transmit. Thus, credit starvation might not actually result in a performance decrease.
- Credit starvation limits maximum throughput by the amount of the congestion. For instance, if you are running at 20 percent congestion, the port will be limited to 80 percent of maximum bandwidth. If you are not running anywhere near that amount of traffic, the only affect will be an immeasurable increase in response time.
- In some environments, a level of credit starvation is completely normal. For instance, if your 8 Gb cluster is constantly attempting to transmit data to a 4 Gb storage array or hosts, you can expect to see a constant level of credit exhaustion as the SVC attempts to transmit data faster than the destination port is physically capable of handling it.

  For GM environments with an FCIP link that is usually *far* slower than SVC line rate, starvation percentages up to 75 percent are not unheard of in some mirroring-heavy solutions.

A zero-credit condition indicates only one thing: The upstream device or devices had more data to transmit than the downstream link or links are capable of transmitting during a particular short interval of time. These downstream links include any intervening ISLs, or even the link between a remote SVC node and its switch. Figure 13-3 on page 499 illustrates potential congestion points.

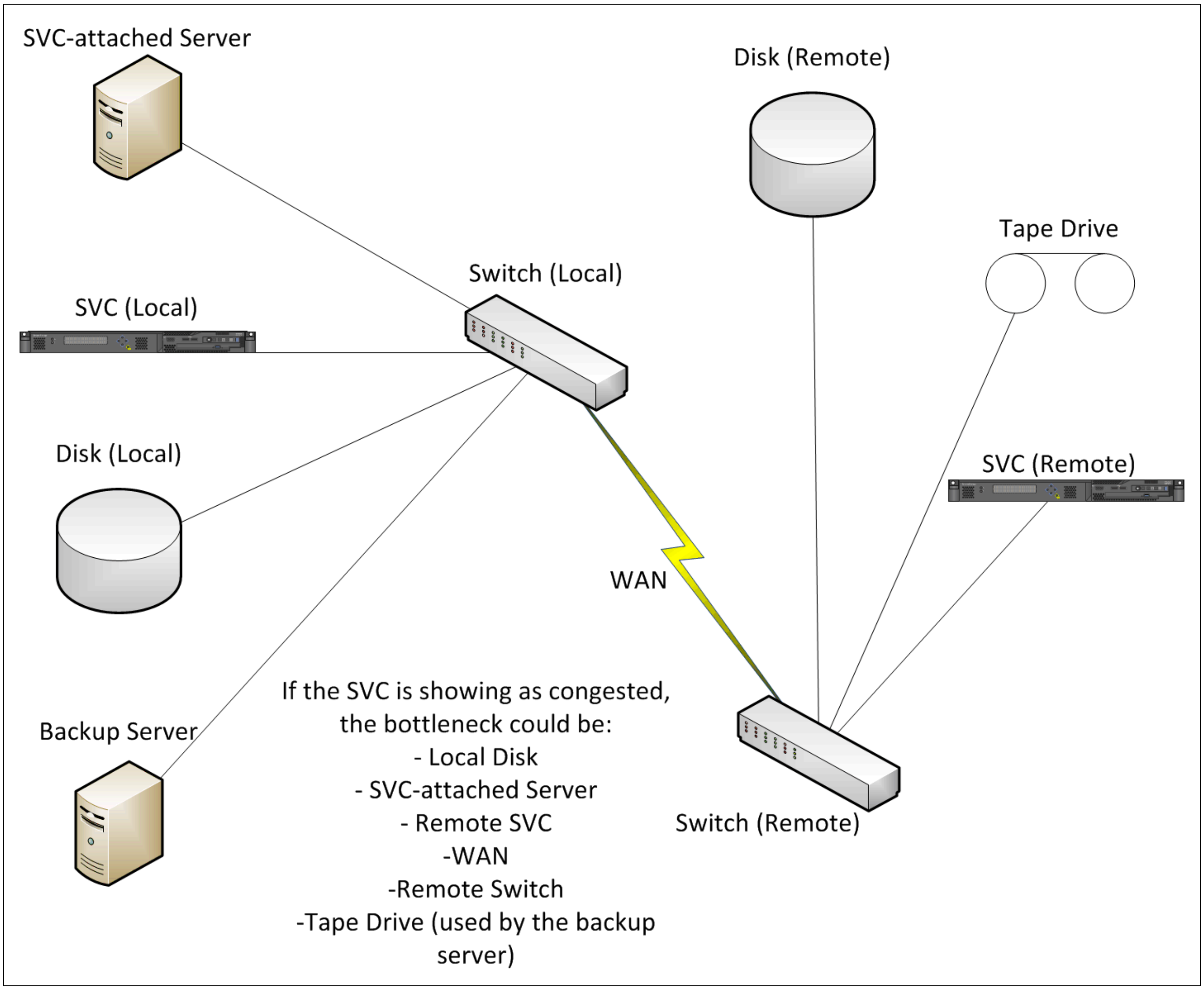


Figure 13-3 Potential congestion points

If an increase in credit starvation is accompanied by an increase in Port to Local Node Response Time, that is a possible indication that the starvation might be affecting your local SVC traffic, in addition to any mirroring performance issues.

As with any performance measurement, buffer credit measurements are most useful when you have long-term “steady-state” data to draw on. If you routinely reach 40 percent credit starvation during your peak workload periods, and it suddenly hits 70 percent during which you also experience your Global Mirror posting an error message, you now have a valuable datapoint. You can then go investigate why the SAN is “pushing back” onto the SVC. The most common cause in a Global Mirror environment is the WAN link.

Alternatively, if you have no previous performance information and your only data point is 70 percent credit starvation, it might lead you to investigate WAN performance when in fact 70 percent starvation is the normal state of your solution, and the true problem lies elsewhere.

FCoE flow control

In Fibre Channel over Ethernet environments, flow control is similar in effect to buffer credit flow control. The only difference is that instead of the receiver acknowledging receipt of each frame with an R_Rdy, an overloaded receiver sends an Ethernet Pause frame, which informs the sender to temporarily cease FCoE data transmission.

As with buffer credit flow control, if a port is Paused, it cannot send any FCoE frames to any destination, even if those other destinations are not congested.

Implications for SVC/V7000 Mirroring

If the SAN is congested, and traffic is throttled due to either of the two preceding mechanisms, the congested port cannot be used for anything until it receives permission to again transmit. That means that in addition to not being able to push traffic to the partner cluster, the port cannot send frames to storage, hosts, or local nodes.

A way around this is the use of mirroring-only ports, as discussed in Chapter 12, “Fabric design considerations for Replication Family” on page 473.

13.4 FCIP congestion troubleshooting

If you utilize FCIP in your mirroring solution, you must ensure it is properly configured and stable; otherwise, your SVC/Storwize V7000 solution might not run in an optimal manner.

FCIP congestion is caused by one of two issues:

- An overconfigured or poor quality FCIP tunnel
- Driving more storage traffic than the FCIP tunnel is capable of handling

13.4.1 Background

It is common for an FCIP tunnel to be configured for more bandwidth than the WAN link is actually capable of handling. When this occurs, the performance of the mirror, and possibly, the SVC/Storwize V7000 cluster, will be poor. 1720 and 1920 errors are common.

The first action many storage administrators take is to consult with their networking team. Field experience has shown that almost invariably the network team will respond with something like: “The link cannot possibly be overloaded; our monitoring shows the link only X% utilized.” (X is usually some number no greater than 20 percent.)

How can the link be using more than the available bandwidth *and* have such low utilization levels? Actually, this is not a contradiction.

The reason why link utilization might appear to be low is because of sampling error. Network monitoring tools operate on a fairly large sampling interval compared to how FCIP and TCP/IP adjust bandwidth. When a link is not performing well, and yet is only 30 percent utilized, the actual bandwidth used looks similar to the graph shown in Figure 13-4 on page 501.

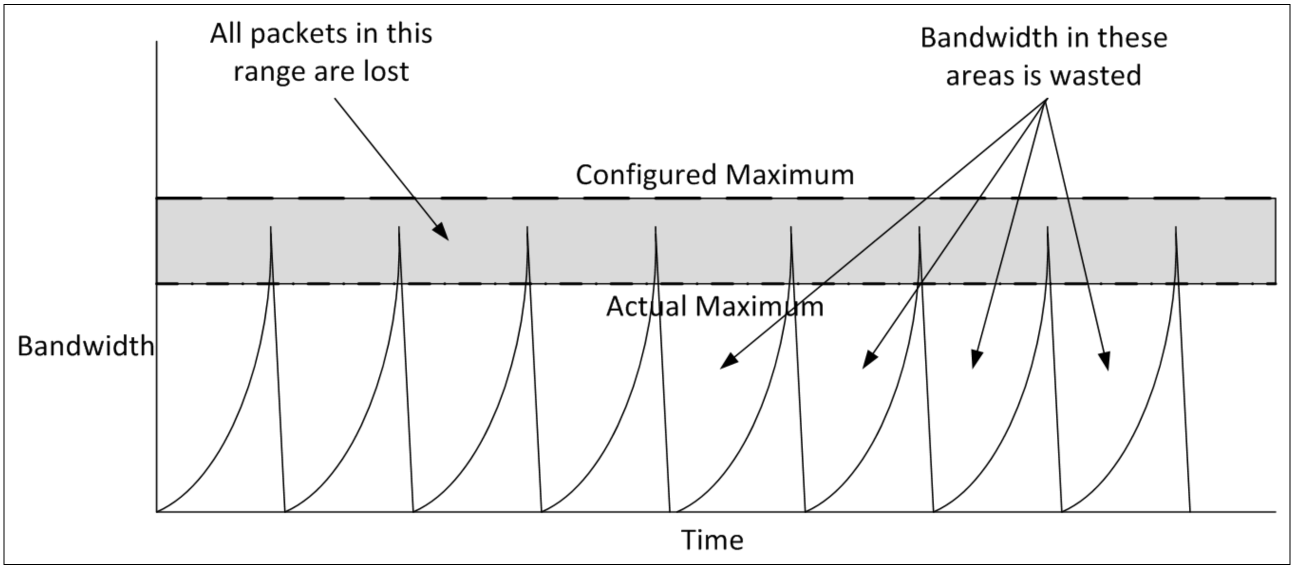


Figure 13-4   FCIP performance during an overload condition

What is happening is that when the used bandwidth surpasses the available bandwidth, retransmissions occur and the throughput immediately drops to zero. It takes some time for both FCIP and TCP/IP to ramp back up until the bandwidth again hits the limit, triggers a re-transmit, and crashes again. This “sawtooth” pattern, which might occur many times every second, severely limits how much bandwidth is actually usable when retransmissions are occurring. As mentioned elsewhere in this book, links become marginal at approximately 0.02 percent re-transmissions, and are usually utterly unusable at a re-transmission level of 0.2 percent.

13.4.2 Checking WAN bandwidth configuration

TCP/IP congestion control is performed through the “sliding window” protocol. In other words, this throttles bandwidth by gradually increasing traffic load until packet loss is encountered. At that point, it “backs off” so the congestion can clear. Unfortunately, this process introduces significant latency and the SCSI protocols used by all SAN storage experience a significant performance impact when this occurs.

Your first step when checking on the FCIP tunnel is to ensure that the bandwidth configured on the FCIP tunnel is not in excess of the *guaranteed* bandwidth available on the WAN link.

Brocade FCIP environment bandwidth configuration check

There are two different FCIP products within the Brocade product line, and they use different configuration commands.

> **Tip:** All bandwidth figures for FCIP configuration and monitoring are displayed by the switch in Kb (kilo*bits*) per second. To convert this into MBps (Mega*bytes*), divide the Kbps by 80.

Brocade 7500, IBM SAN 18B-R Router, or a FR4-18i Director Blade

Use the command **portshow fciptunnel [slot/]<ge_port> al**. Verify that the value displayed under committed bandwidth is no greater than the amount you are guaranteed at all times on your WAN link. Sample output from this command is shown in Example 13-1 on page 502.

If you have multiple tunnels sharing the same WAN link (for instance, tunnels on other SAN routers or GE ports), ensure that you add their bandwidth together.

Example 13-1 Sample portshow fciptunnel command output

```
switch0:admin> portshow fciptunnel ge0 all
Port: ge0
-------------------------------------------
Tunnel ID 0
Tunnel Description Not Configured
Remote IP Addr 10.10.12.100
Local IP Addr 10.62.0.100
Remote WWN Not Configured
Local WWN 10:00:00:05:1e:38:58:61
Compression on
Fastwrite on
Tape Pipelining on
Committed Rate 1000000 Kbps (1.000000 Gbps)
SACK on
Min Retransmit Time 100
Keepalive Timeout 90
Max Retransmissions 9
VC QoS Mapping on
DSCP Marking (Control): 45, DSCP Marking (Data): 30
VLAN Tagging Not Configured
TCP Byte Streaming off
Status : Inactive
Connected Count: 0
IKE Policy 1
IPSec Policy 1
Pre-Shared Key qbcdefghijklmnopqrstuvwxyz123456
```

In Example 13-1, the link is configured for full 1 Gb line rate. If the true bandwidth of the WAN link is slower than this value, reconfigure the tunnel for the correct bandwidth using the command **portCfg fciptunnel [slot/]<ge_port> modify <tunnel_id> -b <rate_in_Kbps>**.

For example, **portcfg fciptunnel ge0 0 -b 150000** would change ge0, tunnel 0, to a committed rate of approximately 150 Mbps, which is near the available payload bandwidth of an OC-3 WAN link.

Brocade 7800, IBM SAN06B-R, Director FX8-24 Blade

To verify FCIP bandwidth on the Brocade 7800, IBM SAN06B-R and Director FX8-24 Blade, execute the command **portshow fciptunnel [slot/]<ve_port> -c**. Sample output from the command (on a switch with Adaptive Rate Limiting enabled) is shown in Example 13-2.

Example 13-2 Sample portshow fciptunnel output (with Adaptive Rate Limiting)

```
switch:admin> portshow fciptunnel 17 -c
-------------------------------------------
Tunnel ID: 17
```

```
Tunnel Description:
Admin Status: Enabled
Oper Status: Up
Compression: On (Moderate)
Fastwrite: Off
Tape Acceleration: Off
TPerf Option: Off
IPSec: Disabled
Remote WWN: Not Configured
Local WWN: 10:00:00:05:1e:55:59:e9
Peer WWN: 10:00:00:05:1e:55:68:05
Circuit Count: 4
Flags: 0x00000000
FICON: Off
-------------------------------------------
Circuit ID: 17.0
Circuit Num: 0
Admin Status: Enabled
Oper Status: Up
Remote IP: 100.83.0.100
Local IP: 100.80.0.100
Metric: 0
Min Comm Rt: 1000000
Max Comm Rt: 1000000
SACK: On
Min Retrans Time: 100
Max Retransmits: 8
Keepalive Timeout: 10000
Path MTU Disc: 0
VLAN ID: 300
L2CoS: F: 7 H: 5 M: 3 L: 1
DSCP: F: 32 H: 16 M: 8 L: 4
Flags: 0x00000000
```

If this switch did not have Adaptive Rate Limiting in use, you would see only a single Committed Rate listed.

If you have multiple circuits that share the same WAN link on the same SAN Router, or on other SAN routers, make sure to add the rates together to ensure you do not exceed the capacity of the link.

In Example 13-2 on page 502, the circuit is set to a Minimum Committed Rate of 1,000,000 Kbps, or 1 Gbps. If the WAN link is different from what is configured, and Adaptive Rate Limiting is in use, correct the bandwidth with the command **portCfg fcipcircuit [slot]<ve_port> modify <circuit_id> -b <rate_in_Kbps> -B <rate_in_Kbps>**.

For instance, to change the circuit in Example 13-2 on page 502 to a guaranteed bandwidth level of 300 Mbps, and a maximum of 500 Mbps, use the command **portCfg fcipCircuit 17 modify 0 -b 300000 -B 500000**.

If Adaptive Rate Limiting is not in use, you must delete and re-create the circuit to modify the committed rate.

Cisco FCIP bandwidth configuration

To verify FCIP bandwidth in a Cisco environment, use the command **`show fcip profile all`**. Sample output from the command is shown in Example 13-3.

Example 13-3 Cisco show fcip profile all output

```
sc9509b-with-Sup2s# show fcip profile all
FCIP Profile 10
    Internet Address is 1.2.3.4  (interface GigabitEthernet3/1)
    Listen Port is 3225
    TCP parameters
        SACK is enabled
        PMTU discovery is enabled, reset timeout is 3600 sec
        Keep alive is 60 sec
        Minimum retransmission timeout is 200 ms
        Maximum number of re-transmissions is 4
        Send buffer size is 0 KB
        Maximum allowed bandwidth is 750000 kbps
        Minimum available bandwidth is 750000 kbps
        Configured round trip time is 10000 usec
        Congestion window monitoring is enabled, burst size is 50 KB
        Auto jitter detection is enabled
```

In Example 13-3, the link is configured for 750 Mbps. To adjust this to a different value, use the command sequence **`config terminal`** → **`fcip profile <profile ID>`** → **`tcp max-bandwidth-mbps <maximum bandwidth in Mbps> min-available-bandwidth-Mbps <minimum bandwidth in Mbps> round-trip-time-ms <round trip delay in ms>`**. You cannot execute this command without inputting the maximum bandwidth, minimum bandwidth, and estimated round trip time.

To modify the profile in Example 13-3 to a minimum bandwidth of 300 Mbps and a maximum of 500 Mbps, use the commands shown in Example 13-4.

Example 13-4 Cisco FCIP bandwidth modification

```
sc9509b-with-Sup2s# config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
sc9509b-with-Sup2s(config)# fcip profile 10
sc9509b-with-Sup2s(config-profile)# tcp max-bandwidth-mbps 500
min-available-bandwidth-mbps 300 round-trip-time-ms 10
sc9509b-with-Sup2s(config-profile)# show fcip profile 10
FCIP Profile 10
    Internet Address is 1.2.3.4  (interface GigabitEthernet3/1)
    Listen Port is 3225
    TCP parameters
        SACK is enabled
        PMTU discovery is enabled, reset timeout is 3600 sec
        Keep alive is 60 sec
        Minimum retransmission timeout is 200 ms
        Maximum number of re-transmissions is 4
        Send buffer size is 0 KB
        Maximum allowed bandwidth is 500000 kbps
        Minimum available bandwidth is 300000 kbps
        Configured round trip time is 10000 usec
        Congestion window monitoring is enabled, burst size is 50 KB
```

```
        Auto jitter detection is enabled
```

13.4.3 FCIP tunnel quality check

As discussed in 13.4.1, "Background" on page 500, retransmits are a key part of finding a problem within your FCIP tunnel. In the presence of significant retransmits, your FCIP tunnel will perform poorly.

One of the most frustrating things about retransmits in an FCIP tunnel is that unless the network link is completely broken, the FCIP tunnel will not actually go down. If it did go down, this would quickly be detected by the SVC/Storwize V7000, mirroring would immediately halt, and little to no performance impact would occur. What happens instead is that the FCIP tunnel stays up, even if it is completely unusable. The tunnel will stay up as long as a single packet is not retransmitted several times. In actual customer installations, retransmission rates of over 300 percent have been observed without the FCIP tunnel going down. A tunnel with such a high retransmission rate cannot accomplish useful work.

In the long term, retransmissions may be monitored using switch monitoring utilities. For more information about the use of those utilities, consult the documentation and the tool vendor.

Using retransmission statistics

Most SAN Routers present cumulative statistics over the life of the FCIP tunnel. If you have not pulled statistics regularly, these numbers are of limited use (for instance, the counters might have wrapped). For this reason, it is useful to retrieve the retransmit counters at least once a week, although once a day is better. (This can be accomplished using a scripting utility such as Expect, along with a task scheduler such as cron.)

If you are in the middle of a performance issue, pull two samples five minutes apart.

Over any interval five minutes to an hour, the maximum retransmission rate where you can be fairly certain it will not cause performance difficulties is 0.02 percent. If the retransmission rate is 0.2 percent or higher, a poor or overload link is almost certainly the cause of your mirroring performance issue.

> **Tip:** When calculating retransmission percentages, hundredths of a percent are often expressed by the calculator in scientific notation. To help you translate this into a more useful number, note that $1x10^{-4}$ is 0.01 percent.
>
> For example, if retransmits divided by total packets returns $2.37x10^{-4}$, then your retransmit percentage is approximately 0.024 percent (this is a marginal value.)

Brocade 7500, IBM SAN 18B-R Router, or a FR4-18i Director Blade

If you are actively having a performance issue, this particular router architecture helpfully provides a 30-second moving average of the retransmission counts. To retrieve these statistics, use the command **`portshow fciptunnel [slot/]<ge_port> all -perf`**. Sample output is shown in Example 13-5. The exact output might vary by Fabric OS version, but the crucial data highlighted in the example is present in all versions.

Example 13-5 Retransmit Count Check for 7500, SAN18B-R and FR4-18i

```
portshow fciptunnel ge1 all -perf:

Port: ge1
-------------------------------------------
```

```
Tunnel ID 0
Tunnel Description Not Configured
Remote IP Addr 1.2.3.4
Local IP Addr 5.6.7.8
Remote WWN Not Configured
Local WWN 10:00:00:05:1e:12:34:56
Compression off
Fastwrite off
Tape Pipelining off
Committed Rate 200000 Kbps (0.200000 Gbps)
SACK on
Min Retransmit Time 100
Keepalive Timeout 10
Max Retransmissions 8
   Status : Active
Connected Count: 10
FC control traffic TCP connection:
    Local 1.2.3.4:4133, Remote 5.6.7.8:3225
    Performance stats:
        1771244 output packets
           0 pkt/s 30s avg, 0 pkt/s lifetime avg
        76097772 output Bytes
           37 Bps 30s avg, 34 Bps lifetime avg
        73 packets lost (retransmits)
           1.00% loss rate 30s avg
        1812716 input packets
           0 pkt/s 30s avg, 0 pkt/s lifetime avg
        77398212 input Bytes
           36 Bps 30s avg, 35 Bps lifetime avg
 Data transfer TCP connection:
    Local 1.2.3.4:4134, Remote 5.6.7.8:3226
    Performance stats:
        12160048721 output packets
           9745 pkt/s 30s avg, 5524 pkt/s lifetime avg
        10185980200472 output Bytes
           8444468 Bps 30s avg, 4628021 Bps lifetime avg
        2889665 packets lost (retransmits)
           1.13% loss rate 30s avg
        8092095614 input packets
           6808 pkt/s 30s avg, 3676 pkt/s lifetime avg
        1021615896236 input Bytes
           915131 Bps 30s avg, 464173 Bps lifetime avg
```

Although many statistics are displayed in the output of Example 13-5 on page 505, the most important are the three lines that appear in bold: the total output packets, the packets lost, and the running retransmit average of the Data transfer connection (not the Control traffic connection.) In the example, the cumulative retransmit percentage over the tunnel uptime is 0.024 percent (2889665 retransmits / 12160048721 total packets). This particular sample provides an excellent example of why the numbers over the life of the tunnel are not always trustworthy, because the switch is also informing that the 30s running average loss rate is 1.13 percent, a value that is well outside what is acceptable for a mirroring solution.

Brocade 7800, IBM SAN06B-R, Director FX8-24 Blade

To retrieve your tunnel statistics, use the command portshow **fciptunnel all -p**. Because this command only provides statistics over the lifetime of the tunnel, to obtain valid data you need to have samples from multiple points in time. Your samples should be at least five minutes apart and include the time of day you are having a problem and/or the highest traffic levels you experience. A sample of the output provided by this command is shown in Example 13-6.

Example 13-6 Sample fciptunnel all -p output

```
portshow fciptunnel all -p :
-------------------------------------------
Tunnel ID: 16
   Tunnel Description:
   Admin Status: Enabled
   Oper Status: Up
   Compression: On (Standard)
   Fastwrite: Off
   Tape Acceleration: Off
   TPerf Option: Off
   IPSec: Disabled
   IPSec Key: ''
   Remote WWN: Not Configured
   Local WWN: 10:00:00:05:33:91:a2:d2
   Peer WWN: 10:00:00:05:33:55:29:9f
   Circuit Count: 1
   Flags: 0x00000000
   FICON: Off
   Oper Status: Up
   Flow Ctrl State: Off
   Connected Count: 39
   Tunnel Duration: 7 days, 8 hours, 7 minutes, 13 seconds
      Compression Statistics:
         1774699512728 Uncompressed Bytes
         729373893832 Compressed Bytes
           2.43 : 1 Compression Ratio
      Performance Statistics Overall Throughput
         Oper Status: Up
         Flow Ctrl State: Off
         Connected Count: 30
         Duration: 7 days, 8 hours, 7 minutes, 16 seconds
         8047813020 Output Bytes
            1586346 Bps 30s Avg, 1269211 Bps Lifetime Avg
         13308202 Output Packets
            26 pkt/s 30s Avg, 2098 pkt/s Lifetime Avg
         191095702780 Input Bytes
            423527 Bps 30s Avg, 301395 Bps Lifetime Avg
         11919760 Input Packets
            2461 pkt/s 30s Avg, 1885 pkt/s Lifetime Avg
         TCP Stats:
            8700928828 Output Bytes
            16218953 Output Packets
            1916381104 Input Bytes
            13410752 Input Packets
            Retransmits: 4580
            Round Trip Time: 38 ms
```

```
        Out Of Order: 3352
        Slow Starts: 401
```

Look at the two lines highlighted in bold in the example to calculate the retransmit percentage. In this example, the retransmit percentage is 0.028 percent, which is slightly higher than what is considered to be an acceptable value. If you see a retransmit percentage like this in your environment, it is useful to take further measurements at shorter sampling intervals to see if it rises much higher during any particular time of day, indicating that the load might be too high at those points.

Cisco FCIP solutions

To examine FCIP retransmit statistics on a Cisco FCIP connection, use the command **show ips stats tcp all**. Sample output is shown in Example 13-7.

Example 13-7 Cisco show ips stats tcp all output

```
TCP Statistics for port GigabitEthernet6/1
    TCP send stats
      33163293665 segments, 13157630253546 bytes
      25694154111 data, 6994721980 ack only packets
      10323 control (SYN/FIN/RST), 3 probes, 1640778 window updates
      472769196 segments retransmitted, 654770858509 bytes
      65843058 retransmitted while on ethernet send queue, 277955104 packets split
      26619831 delayed acks sent
    TCP receive stats
      25754305093 segments, 15257016494 data packets in sequence, 1691709278480
bytes in sequence
      10603160235 predicted ack, 15017178790 predicted data
      0 bad checksum, 0 multi/broadcast, 0 bad offset
      0 no memory drops, 0 short segments
      555018876 duplicate bytes, 4459298 duplicate packets
      0 partial duplicate bytes, 0 partial duplicate packets
      373004980 out-of-order bytes, 3217922 out-of-order packets
      27 packet after window, 0 bytes after window
      790 packets after close
      10984849517 acks, 13020235134304 ack bytes, 5 ack toomuch, 3074923 duplicate
acks
      1 ack packets left of snd_una, 53 non-4 byte aligned packets
      1363317 window updates, 0 window probe
      13333 pcb hash miss, 6430 no port, 1053 bad SYN, 6 paws drops
    TCP Connection Stats
      1607 attempts, 2398 accepts, 1721 established
      5169 closed, 662 drops, 307 conn drops
      686 drop in retransmit timeout, 1 drop in keepalive timeout
      0 drop in persist drops, 0 connections drained
    TCP Miscellaneous Stats
      551181713 segments timed, 9191688023 rtt updated
      3634 retransmit timeout, 22 persist timeout
      70367 keepalive timeout, 70366 keepalive probes
    TCP SACK Stats
      0 recovery episodes, 20157367721 data packets, 14957707353644 data bytes
      472549896 data packets retransmitted, 654472575952 data bytes retransmitted
      464 connections closed, 238158 retransmit timeouts
    TCP SYN Cache Stats
      2711 entries, 2398 connections completed, 7 entries timed out
```

```
0 dropped due to overflow, 9 dropped due to RST
0 dropped due to ICMP unreach, 0 dropped due to bucket overflow
0 abort due to no memory, 707 duplicate SYN, 0 no-route SYN drop
0 hash collisions, 261 retransmitted
```

In the output displayed in Example 13-7 on page 508, the important statistics are highlighted in bold. This connection is showing a retransmit percentage of 1.8 percent (472769196 data packets / 25694154111 total segments.) This is far in excess of what any mirroring solution will tolerate. You can reset these statistics with the command **`clear ips stats all`**.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

The following IBM Redbooks publications provide additional information about the topic in this document. Note that some publications referenced in this list might be available in softcopy only.

- *Implementing the IBM System Storage SAN Volume Controller V6.3*, SG24-7933
- *Implementing the IBM Storwize V7000 V6.3*, SG24-7938
- *SAN Volume Controller: Best Practices and Performance Guidelines*, SG24-7521
- *IBM/Cisco Multiprotocol Routing: An Introduction and Implementation*, SG24-7543
- *IBM System Storage b-type Multiprotocol Routing: An Introduction and Implementation*, SG24-7544
- *Implementing an IBM/Brocade SAN with 8 Gbps Directors and Switches*, SG24-6116
- *Implementing the Storwize V7000 and the IBM System Storage SAN32B-E4 Encryption Switch*, SG24-7977
- *Implementing the IBM Storwize V7000 V6.3*, SG24-7938
- *IBM TotalStorage Productivity Center for Replication Using DS8000*, SG24-7596
- *IBM SAN Volume Controller and Brocade Disaster Recovery Solutions for VMware*, REDP-4626
- *IBM System Storage SAN Volume Controller Upgrade Path from Version 4.3.1 to 6.1*, REDP-4716
- *Real-time Compression in SAN Volume Controller and Storwize V7000*, REDP-4859

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:

- *IBM System Storage Open Software Family SAN Volume Controller: Planning Guide*, GA22-1052
- *IBM System Storage Master Console: Installation and User's Guide*, GC30-4090
- *IBM System Storage Open Software Family SAN Volume Controller: Installation Guide*, SC26-7541
- *IBM System Storage Open Software Family SAN Volume Controller: Service Guide*, SC26-7542

- *IBM System Storage Open Software Family SAN Volume Controller: Configuration Guide*, SC26-7543
- *IBM System Storage Open Software Family SAN Volume Controller: Command-Line Interface User's Guide*, SC26-7544
- *IBM System Storage Open Software Family SAN Volume Controller: CIM Agent Developers Reference*, SC26-7545
- *IBM TotalStorage Multipath Subsystem Device Driver User's Guide*, SC30-4096
- *IBM System Storage Open Software Family SAN Volume Controller: Host Attachment Guide*, SC26-7563
- *IBM System Storage SAN Volume Controller and IBM Storwize V7000, Version 6.3.0: Command-Line Interface User's Guide*, GC27-2287
- *IBM System Storage SAN Volume Controller, Version 6.3.0, CIM Agent Developer's Guide,* GC27-2288
- *IBM System Storage SAN Volume Controller CIM Agent Developer's Reference,* SC26-7904

Referenced websites

These websites are also relevant as further information sources:

- IBM System Storage home page

  http://www.storage.ibm.com
- SAN Volume Controller supported platform

  http://www-1.ibm.com/servers/storage/support/software/sanvc/index.html
- Download site for Windows SSH freeware

  http://www.chiark.greenend.org.uk/~sgtatham/putty
- IBM site to download SSH for AIX

  http://oss.software.ibm.com/developerworks/projects/openssh
- Open source site for SSH for Windows and Mac

  http://www.openssh.com/windows.html
- Cygwin Linux-like environment for Windows

  http://www.cygwin.com
- IBM Tivoli Storage Area Network Manager site

  http://www-306.ibm.com/software/sysmgmt/products/support/IBMTivoliStorageAreaNetworkManager.html
- Microsoft Knowledge Base Article 131658

  http://support.microsoft.com/support/kb/articles/Q131/6/58.asp
- Microsoft Knowledge Base Article 149927

  http://support.microsoft.com/support/kb/articles/Q149/9/27.asp
- Sysinternals home page

  http://www.sysinternals.com

- Subsystem Device Driver download site

  http://www-1.ibm.com/servers/storage/support/software/sdd/index.html

- IBM TotalStorage Virtualization home page

  http://www-1.ibm.com/servers/storage/software/virtualization/index.html

Further reading for disaster recovery:

- Interdata center workload mobility and failover with VMware vSphere and SAN Volume Controller Stretched Cluster
  http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101923
- Disaster recovery for SAP with VMware Site Recovery Manager on IBM Storwize V7000
  http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101913
- SVC Global Mirror - a practical review of important parameters
  http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101848

Further reading about Backup and Restore:

- Storwize V7000 with Tivoli FlashCopy Manager: Backup Solution for Exchange 2010

  http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101764

- Protecting VMware data with Tivoli Storage FlashCopy Manager for VMware and Tivoli Storage Manager for Virtual Environments

  http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP102021

- SAP with IBM Tivoli Storage FlashCopy Manager for VMware, XIV and Storwize V7000

  http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP102093

- Protecting Oracle RAC ASM databases with IBM Tivoli Storage FlashCopy Manager

  http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP102113

- Storwize V7000 Practice guide: Backup & restore of Oracle Database using Tivoli FlashCopy Manager

  http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101771

- SVC 6.3 Copy Services for Backup & Recovery of Oracle 11.2 RAC/ASM Databases

  http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP102080

- Using Symantec NetBackup and IBM Storwize V7000 FlashCopy/Concepts and best practices for backup and restore solution on Linux

  http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101767

- Epic and Storwize V7000 - Solution overview and performance benchmark with FlashCopy

  http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP102011

Further reading about TPC-R:

- IBM Tivoli Storage Productivity Center Information Center

  http://publib.boulder.ibm.com/infocenter/tivihelp/v4r1/index.jsp?topic=%2Fcom.ibm.help.ic.doc%2Fusing_system%2Fabout_the_system.html

- IBM Redbooks publications about TPC-R

  http://www.redbooks.ibm.com/cgi-bin/searchsite.cgi?query=tpc+AND+replication

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

IBM

Redbooks

IBM System Storage SAN Volume Controller and Storwize V7000 Replication Family Services

(1.0" spine)
0.875"<->1.498"
460 <-> 788 pages

IBM System Storage SAN Volume Controller and Storwize V7000 Replication Family Services

Describes new Replication Family Services functionality

Provides server integration examples

Includes scripting examples

This IBM Redbooks publication describes the new features that have been added with the release of the IBM System Storage SAN Volume Controller (SVC) and IBM System Storage Storwize V7000 6.4.0 code, including Replication Family Services.

Replication Family Services refers to the various copy services available on the SVC and Storwize V7000 including IBM FlashCopy, Metro Mirror and Global Mirror, Global Mirror with Change Volumes, Volume Mirroring, and Stretched Cluster Volume Mirroring. The details behind the theory and practice of these services are examined, and SAN design suggestions and troubleshooting tips are provided. Planning requirements, automating copy services processed, and fabric design are explained. Multiple examples including implementation and server integration are included, along with a discussion of software solutions and services that are based on Replication Family Services.

This book is intended for use by pre-sales and post-sales support, and by storage administrators. Readers are expected to have advanced knowledge of the SVC and Storwize V7000, and of the SAN environment. A list of publications that are useful resources for background information about these topics is provided for your reference.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-7574-02 ISBN 0738437786