EMC® Solutions Enabler Array Management

Version 8.0.1

CLI User Guide

REV 01

EMC²

Copyright © 2014 EMC Corporation. All rights reserved. Published in the USA.

Published November, 2014

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

The information in this publication is provided as is. EMC Corporation makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose. Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

EMC², EMC, and the EMC logo are registered trademarks or trademarks of EMC Corporation in the United States and other countries. All other trademarks used herein are the property of their respective owners.

For the most up-to-date regulatory document for your product line, go to the technical documentation and advisories section on the EMC online support website.

CONTENTS

Preface

Part 1 **Setting up and Configuring Storage Arrays**

Chapter 1 **Introduction**

Overview 28
Getting SYMCLI help 29
Top level command help 29
Individual command help 29
Command input tips 29
Time-saving tips 29
Viewing truncated output 30
Referencing devices and objects 30
Environment variables 30
Presetting names and IDs 31
Setting group names to display full names 31
Setting compatibility 31
Prior Solutions Enabler scripts 31
Compatibility for CLI displays 32
Client/server compatibility restrictions 32

Chapter 2 **Discovery**

Overview 36
Verifying host configuration database is synchronized 36
Scanning for new devices 37
Authorizing connectivity 37
Configuring virtual disk mapping 38
Working with the host configuration database file 39
Database file location 39
Database file lock 39
Changing the current database file 39
Changing the database in client/server mode 40
Database access modes 40
Command modes: Online and offline 40
Database configuration information 41
Synchronizing configuration data 41
Sync operations 41
Inhibiting database synchronization 42

Chapter 3 **User and Group Authorization**

Overview 44
User roles 44
Multiple user roles 45
Authorization rules 45
User ID formats 46
Managing authorization settings 47
Identifying the current username 47

Setting up authorization 48
Enabling and disabling authorization 49
Setting enforcement 49
Modifying role mappings 49
Command file usage 50
Backing up and restoring the authorization database 51
Monitoring authorization 52
Listing roles, names, and components 52

Chapter 4 Host-based Access Control

Overview 56
Access Control database 56
Securely identifying hosts with access IDs 57
Creating and modifying access control data 58
Command file execution 58
Minimum access control configuration 59
Initial ACL setup 59
Creating and managing access groups 60
Setting up an access group 60
User access IDs (PINs) for the AdminGrp 61
Editing and managing an access group 62
Creating and managing limited access to access pools 63
Setting up access pools 63
Editing and managing access pools 64
Creating and managing access control entries 65
Available permissions 65
Obtaining access control information 69
Releasing pending locked sessions 70
Understanding access strategies 71
Initially grant all permissions to everybody 71
Customizing, granting and denying default access 71
BCV setup strategy 73
TimeFinder/Snap setup strategy (Enginuity 5876 or lower) 74
SRDF setup strategy 74
RecoverPoint Splitter setup strategy (Enginuity 5876 or lower) 74
FAST/Optimizer setup strategy (Enginuity 5876 or lower) 75
Symmetrix Management Console/Unisphere setup strategy 75
Symaccess setup strategy 75
Backing up and restoring the access control database 75
Create a backup file 75
Restore the backup file to the access control database 75

Chapter 5 Grouping Devices

Overview 78
Device groups 78
Group name services 78
Names of device groups and devices 79
Device group types 79
Device lists 79
Creating a device group 80
Device restrictions 80
Adding devices to a device group 82
Listing device groups 85

Showing details about a device group ... 86
Exporting and importing device lists ... 86
Renaming device groups ... 88
Moving devices ... 88
Copying devices ... 89
Removing devices ... 90
Deleting a device group ... 91
Performing operations on device group devices ... 91
Storage groups ... 93
Restrictions ... 93
Working with storage groups ... 94
Setting Host I/O Limits for storage groups ... 104
Modifying the host ID of devices in a storage group (HYPERMAX OS 5977) . 108
Copying devices from a storage group to a device group ... 109
Listing storage groups ... 109
Showing details about a storage group ... 112
Exporting and importing device lists ... 114
Renaming, moving, copying, and deleting storage groups ... 117
Performing miscellaneous operations on storage group devices ... 122
Composite groups ... 124
Device members ... 124
Logical device name support ... 124
Device group members ... 124
Composite group types ... 126
SRDF consistency groups ... 126
Creating a composite group ... 127
Exporting a composite group to file ... 128
Deleting a composite group ... 129
Importing a composite group ... 129
Adding devices to a composite group ... 129
Adding and removing device groups ... 130
Moving and copying devices ... 131
Removing standard devices ... 132
Renaming a composite group ... 133
Setting SRDF group attributes ... 133
Composite group creation and output example ... 133
Displaying composite group information ... 134
Showing composite groups associated with a device group ... 136
Performing miscellaneous operations on composite group devices ... 136
Using the GNS repository ... 138
Automatic upgrade ... 139
Shared group definitions with GNS ... 139
How GNS handles updates to group definitions ... 141
Configuring GNS ... 143
GNS daemon control ... 144
Mirroring remote device group definitions ... 148
Identifying GNS groups ... 150
GNS backup and restore mechanism ... 151
Troubleshooting GNS ... 152

Chapter 6 Fully Automated Storage Tiering

Fully Automated Storage Tiering (HYPERMAX OS 5977) ... 156
Modifying Storage Resource Pools ... 157

Reporting information 158
Fully Automated Storage Tiering (Enginuity 5874 to 5876) 166
FAST configuration 167
Managing tiers 168
Managing policies 176
Managing storage groups 181
Managing the FAST controller 186
Displaying FAST information 196
FAST reports 197

Chapter 7 Managing Configuration Changes

Introduction 204
Configuration change session rules 204
Symconfigure command execution formats and options 204
Verifying and checking sessions 206
Making configuration changes safely 207
Mixed array environments 208
Supported configuration operations 209
Setting array attributes 213
Managing external disk groups 214
Creating devices (HYPERMAX OS 5977) 214
Restrictions when setting device names 216
Restrictions when creating devices 216
Creating devices (Enginuity 5773 through 5876) 217
Restrictions when setting device names 224
Restrictions when creating devices 225
Copying devices 226
Managing metadevices (Enginuity 5773 through 5876) 228
Configuring thin metadevices (Enginuity 5773 through 5876) 235
Converting devices 237
Setting device attributes 241
Setting device geometry (Enginuity 5773 through 5876) 243
Setting device identifiers 246
Deleting devices 247
Restrictions (HYPERMAX OS 5977) 248
Restrictions (Enginuity 5773 through 5876) 248
Managing device reservations 249
Managing device pools (HYPERMAX OS 5977) 251
Managing device pools (Enginuity 5773 through 5876) 251
Device pool attributes and rules 252
Creating a device pool 253
Deleting a device pool 253
Adding devices to a pool 253
Removing devices from a pool 253
Enabling/disabling devices 254
Renaming a device pool 254
Monitoring a device pool 255
Viewing a device pool 255
Mapping devices 256
Mapping CKD devices (Enginuity 5773 through 5876) 257
Mapping a range of devices 257
Copying the mapping of a range of devices 258
Assigning PAV alias addresses to mapped devices 258
Removing PAV alias addresses from mapped devices 259

Unmapping devices 259
Introducing devices to a host 261
Sun Solaris systems 261
HP-UX systems 261
IBM AIX systems 261
HP Tru64 UNIX systems 262
Windows systems 262
Configuring dynamic RDF-capable devices (Enginuity 5773) 262
Enabling dynamic SRDF on the device 263
Setting SRDF group attributes 263
Swapping RA groups 264
Converting directors (Enginuity 5876 only) 264
Port to director emulation support (HYPERMAX OS 5977) 265
Associating ports to director emulations 265
Disassociating ports from director emulations 267
Setting port characteristics 268

Chapter 8 Configuring Virtual Provisioning

Overview 276
Overview for arrays running HYPERMAX OS 5977 276
Overview for arrays running Enginuity 5773 through 5876 277
Creating DATA devices (Enginuity 5773 through 5876) 278
DATA device configuration rules 278
Creating thin devices (Enginuity 5773 through 5876) 279
Example for creating thin devices 280
Thin device configuration rules 280
Creating thin pools (Enginuity 5773 through 5876) 281
Thin pool restrictions and rules 282
Renaming thin pools 283
Setting pool attributes (Enginuity 5773 through 5876) 283
Performing actions on thin devices (Enginuity 5773 through 5876) 284
Binding thin devices 285
Unbinding thin devices 286
Rebinding thin devices 286
Allocating and reclaiming space on thin devices 288
Restrictions 288
Reclaiming allocations 288
Freeing written tracks (Enginuity 5874 through 5876) 289
Freeing all allocations or written tracks (HYPERMAX OS 5977 only) 289
Stopping background operations on thin devices 290
Restrictions for stopping background operations 290
Draining and rebalancing a thin pool (Enginuity 5874 through 5876) 291
Draining rules 291
Restrictions 291
Deactivating devices in a thin pool 291
Starting the drain on DATA devices 292
Activating devices in a thin pool 292
Stopping the drain on DATA devices 292
Starting pool rebalancing 293
Stopping pool rebalancing 294
Automated pool rebalancing 295
Virtual Pool (VP) compression (Enginuity 5876 only) 295
VP compression rules 296
Verifying the pool and device states 296

Monitoring thin pools 297
Viewing thin pools 298

Chapter 9 Federated Tiered Storage

What is FTS? 302
FTS for HYPERMAX OS 5977 302
FTS for Enginuity 5876 302
Using eDisks 303
Encapsulation 303
External provisioning (Enginuity 5876 only) 303
Geometry of encapsulated devices (Enginuity 5876) 304
Geometry limited devices 304
Creating an external disk group (Enginuity 5876 or lower) 306
Restrictions 306
Deleting a disk group (Enginuity 5876 or lower) 306
Restrictions 306
Adding an eDisk 307
Restrictions 307
Removing an eDisk 308
Displaying external disk information 309
Solutions Enabler FTS support (Enginuity 5876 only) 309

Chapter 10 Masking Devices with Auto-Provisioning Groups

Auto-provisioning groups 312
Provisioning limits 312
Creating a masking view 313
Creating an auto-provisioning session 314
Discovering host HBAs 314
Creating groups and views 315
Managing masking views 319
Deleting masking views 319
Naming groups and views 319
Backing up and restoring views 320
Copying groups and views 320
Managing storage groups 321
Adding devices 321
Removing devices 322
Renaming a storage group 322
Deleting storage groups 322
Listing and showing storage group information 323
Managing port groups 323
Adding ports 323
Removing ports 324
Deleting port groups 324
Copying a port group from array to array 324
Listing and showing port group information 324
Locking down a Fibre Channel ID 325
Managing initiator groups 326
Adding initiators 326
Removing initiators 327
Deleting initiator groups 327
Setting initiator group flags 327
Setting HBA flags 329

Replacing an HBA 330
Renaming an HBA 330
Using CHAP authentication 331
Verifying the database 332
Displaying auto-provisioning group information 332
Showing the last update time 333
Displaying masking views 334
Viewing group details 336
Viewing device assignments 338
Listing initiator group devices 339
Viewing the HBA alias name 339

Chapter 11 Masking Devices (Enginuity 5773 or lower)

Overview 342
Host access process 342
Managing access for unsupported host platforms 342
Device masking controls overview 343
Database device locking 343
Supported topologies 344
Fibre Channel topology 344
Native iSCSI topology 344
HBA initiator support 344
Using the device masking database 345
Configuring device masking 345
Access control environment setup 345
Identifying configuration components 346
Initializing and updating the database 347
Enabling authentication 347
iSCSI CHAP authentication: Enginuity version 57xx 347
iSCSI software driver configuration 350
RADIUS server configuration 362
Recommendations for activating the configuration 363
Discovering host HBAs 364
Adding masked devices 365
Removing masked devices 368
Maintaining the device masking database 369
Viewing the VCMDB 372
Managing HBA initiators 374
Managing the Fibre Channel-to-host interface 376

Chapter 12 Managing Quality of Service

Overview 386
Setting copy pace and priority 386
Setting host I/O priority (Enginuity 5876 or lower) 389
Setting bandwidth for Open Replicator (Enginuity 5876 or lower) 390
Setting and resetting LRU parameters 391
Setting RDF director workload distribution 391
Managing dynamic cache partitions (Enginuity 5876 or lower) 393

Chapter 13 Virtual LUN Technology (Enginuity 5874 through 5876)

Overview 400
Migration requirements by features 400

Virtual LUN technology restrictions 401
Virtual LUN VP mobility 402
Device pool-level migrations 402
VP compression guidelines for migration 402
FTS support 403
Creating virtual LUN migration sessions 403
Specifying source devices 403
Specifying target criteria 403
Using files 404
Establishing a migration session 404
Validating a migration session 405
Migrating to configured space 405
Migrating using a device file 405
Migrating using target criteria 406
Error reporting 406
Migrating to unconfigured space 407
Error reporting 408
Migrating to a thin pool 409
Viewing a thin migration 409
Managing virtual LUN migration sessions 410
Listing sessions 411
Querying sessions 411
Verifying sessions 413
Terminating sessions 413
Viewing the audit log 414

Part 2 Querying and Reporting

Chapter 14 Configuration Query Operations

Configuration data 418
SCSI-level data 418
SCSI device information 419
Devices listed by array ID 419
Devices without capacity 419
Devices with WWN 420
Pdevfile format 420
HBA information 420
Mapping information 421
Device identifiers 422
Array-level data 422
Listing arrays 423
Listing array bay locations (Enginuity 5876 or lower) 424
Setting VMAX 10K and 40K LED door feature (Enginuity 5876 only) 424
Listing array bay LED states (Enginuity 5876 or lower) 425
Monitoring Data at Rest Encryption (Enginuity 5875 and 5876) 426
Listing data encryption status 427
Viewing application registrations with array access 429
Listing host connections to arrays 430
Listing director configuration data 431
Array external locks 437
LRU cache management group 439
Optimized Read Miss mode (Enginuity 5876 only) 439
Network services 440
Memory board information 441

Mainframe CU image 441
Enginuity patches 441
Environmental data 442
Device pools 445
Listing feature registrations and usage data 449
Device-level data 449
Device types - HYPERMAX OS 5977 450
Device types - Enginuity 5876 and lower 451
Listing array device information 452
Viewing Recoverpoint Splitter devices (HYPERMAX OS 5977 only) 454
Viewing external spindle devices 454
Device states 456
Viewing device service states 457
Device emulation 457
Showing details about a device 458
Showing clone state flags 459
Showing disk geometry details 460
DATA and SAVE devices 461
Thin device information 462
Migrated devices 467
Pinned devices 469
Filtering device data 469
Releasing device external locks 474
Disk-level data 475
List and show disk information 475

Chapter 15 Statistics

Overview 486
Reporting objects 486
Performance type options 487
REQUESTS statistics 487
Array statistics 488
Device statistics 488
SRDF/A session statistics 488
R-Copy session statistics 489
Mainframe (Ficon) statistics 489
BACKEND statistics and PREFETCH activity 489
Returning BACKEND statistics 490
Prefetch activity 490
Director statistics 490
Director I/O requests 490
R-Copy session statistics 491
PORT statistics 492
How to return director port statistics 492
R-Copy session director port statistics 492
ISCSI statistics (GigE) 492
MEMIO statistics 493
Disk statistics 493
Specifying a disk 494
Returning spindle statistics 494
CACHE statistics 494
Front-end director statistics 494
SRDF/A session statistics 495
PATH statistics 496

CYCLE statistics 497
Example: All device groups 497
RDF statistics 498
SRDF/A session statistics by RA group 499
SRDF device-level statistics 499
SRDF statistics by director number 499
SRDF link-level statistics (Enginuity 5876 or lower) 499

Chapter 16 Change Tracker (Enginuity 5876 or lower)

Overview 502
Practical uses 502
Change Tracker command summary 503
Using -force with delete and remove 504
Monitoring data objects for changes 504
Creating a Change Tracker session 505
Setting the session type 505
Creating a device text file 506
Marking a data object 506
Listing marked objects 507
Removing marked objects 507
Deleting marked objects 508
Viewing marked objects 508
Generating reports of changes 509
SUM reports 509
DELTA reports 510
Viewing reports 511
Report options 511
Example procedure 512
Using Change Tracker for SRDF capacity planning 513

Chapter 17 Optimizing Array Performance (Enginuity 5876 or lower)

Overview 516
Data movement 517
Configuration requirements 517
How does Optimizer work? 518
Performance metrics 519
Modeling service time 519
Determining device swaps 519
Finding the best swap 520
Swap procedure for Enginuity 5874 -5876 520
Swap procedure for Enginuity 5773 and earlier 521
Performing Optimizer operations using symoptmz 523
Enabling and disabling Optimizer 524
Committing command file changes 524
Clearing the Optimizer statistics 524
Setting control parameters 524
Setting advanced parameters 525
Setting time windows 526
Setting the swap priority for devices 526
Setting manual swap lists 526
Enginuity 5874 and 5876 526
Enginuity 5773 and earlier 527
Reviewing Optimizer plans 527

Show parameters 527
Showing control parameters 528
Showing the swap list 529
Approving a swap list 533
Performing rollbacks 534
Reading Optimizer logs 534
Migrating devices 535

Chapter 18 Managing Time Windows (Enginuity 5876 or lower)

Overview 538
Time window restrictions and guidelines 539
Time window weekly format 539
Converting time windows 539
Conversion guidelines 540
Conversion examples 540
Adding time windows 541
Removing time windows 542
Removing all time windows 543
Displaying time windows 543
Using the -summary option 544

Chapter 19 Performing Double Checksum Operations (Enginuity 5876 or lower)

Overview 548
Traditional methods of preventing data corruption 548
Benefits of checking within the array 549
Implementing EMC Double Checksum for Oracle 549
Other checksum operations 549
Suppressing default features 550
Enabling checksum options 551
Verifying checksum is enabled 551
Validating for checksum operations 551
Disabling checksum 552
Functionality 552
Implementing Generic SafeWrite for generic applications 553
Using Generic SafeWrite to protect applications 553
Where to enable Generic SafeWrite 553
Configuring Generic SafeWrite 554
Setting the RDB_cksum device flag 555
Enabling Generic SafeWrite 555
Disabling Generic SafeWrite 556
Listing Generic SafeWrite devices 557
Performance considerations 557
Syntax and examples 557

Chapter 20 Managing Network IPsec (Enginuity 5876 or lower)

Overview 560
IPsec policies 560
IPsec and EMC arrays 561
SYMCLI IPsec management component 561
Viewing policy information 562
Listing policies 562
Showing policy detail 562

Listing policy statistics 564
Command file scripting for creating/modifying policies 565
Command file syntax adding/modifying policies 565
Scripting the policy syntax 566
Scripting proposal set syntax 568
Manual options 569
Proposal options 569
Transform options 569
Committing a policy-scripted command file 571
Preview the file 571
Commit the file 571
Deleting a policy-scripted command file 571

Chapter 21 Events and Logs

Overview 574
Configuring log options 574
Monitoring VMAX events using SYMCLI 576
Monitoring events 576
Listing events 576
Common audit log 576
Example output 577
Event daemon 579

Chapter 22 XML Structured Output

Overview 582
XSLT: XML data transformations 582
Element-based XML 582
Using XML mode 583
Example output 583

Part 3 Migrating Data

Chapter 23 Open Replicator Operations

Open Replicator overview 588
Practical uses 588
Functionality 589
Rcopy concepts 589
Operational rules and limitations 591
Device guidelines 594
SAN setup requirements 594
SYMCLI symsan support 595
Open Replicator command summary 596
Open Replicator operations 597
Hot pull copy session example 597
Cold push copy session example 598
Using Open Replicator to perform a migration 600
Creating a device file 601
Creating a session 603
Activating a session 609
Monitoring session status 611
Terminating a session 617
Removing a remote device from a session 617

Recreating a session 618
Recovering from a failed session 618
Restoring a session 619
Using VDEVs in Open Replicator 620
Open Replicator examples 622
Example 1: Performing a hot pull operation 622
Example 2: Pushing data using BCVs as the control devices 638
Example 3: Pulling data online from IBM F20 to VMAX Family array..... 644
Example 4: Pushing data online from VMAX Family array to Hitachi HDS 9960 array 650
Example 5: Performing a cold push from a VDEV..... 658

Chapter 24 Operational Restrictions and State Reference

Device control operations allowed for ORS control devices 664
ORS operations allowed for replication session states..... 665
TimeFinder/Mirror and TimeFinder/Clone sessions)..... 665
TimeFinder/SnapVX sessions (HYPERMAX OS 5977) 670
SRDF sessions..... 672
ORS operations allowed for device types..... 676

Chapter 25 Federated Live Migration Operations (Enginuity 5876 or lower)

Federated Live Migration overview 680
The FLM process 681
Environmental and system requirements for FLM..... 682
Array/device requirements 682
Array/device limitations 682
Federated Tiered Storage support..... 683
Configuring the application hosts..... 683
Zoning and masking prerequisites 683
Using FLM to perform a migration..... 684
Creating a device file..... 684
Creating a session..... 684
Verifying the created session 685
Activating a FLM session 686
Monitoring FLM session status 687
Failing Back 690
Terminating a FLM session 691
Restoring new array device identity 691
Restoring donor device to host-active mode 691

FIGURES

	Title	Page
1	Access Control Lists and Access Control Entries in the Access Control database	56
2	Example default configuration	71
3	Host-visible GNS state	140
4	GNS across multiple hosts	141
5	Mirrored remote group definition	150
6	Concatenated metadevices	229
7	Striped data	230
8	Masking view overview	313
9	Storage group SG_1	315
10	Port group PG_1	316
11	Initiator group IG_1	317
12	Masking view MV_1	318
13	Device masking components	346
14	A DeltaMark bitmap table	502
15	SRDF configuration	514
16	Optimizer application architecture	516
17	Swap steps using one DRV	521
18	Identify volumes to swap	522
19	Copy volume to DRV	522
20	Copy DRVs to new locations	522
21	DRVs are split	523
22	Control array device pull operation	590
23	Control array device push operation	591
24	Control array hot pull using the symrcopy command	598
25	Control array device cold push using the symrcopy command	599
26	Cold push copy session: single device to multiple remote devices	606
27	Example FLM configuration	681
28	FLM process animation	682

TABLES

	Title	Page
1	SYMCLI help options	29
2	Initial access groups set	59
3	Initial delivered ACLs setup	59
4	Permissions: AccessType	65
5	FAST version differences (Enginuity 5876 and lower)	166
6	Symconfigure command options	205
7	Configuration change operations	209
8	Possible metric names (HYPERMAX OS 5977)	213
9	Possible metric names (Enginuity 5773 through 5876)	213
10	Maximum device sizes (HYPERMAX OS 5977)	215
11	Steps to create a device	217
12	Maximum device sizes (Enginuity 5773 through 5876)	220
13	Metadevice sizes by Enginuity version for auto_meta	229
14	Steps to create an SRDF BCV metadevice	230
15	Metadevice data preservation	235
16	Thin device conversions	239
17	SCSI protocol port flags	269
18	Fibre protocol port flags	271
19	Supported thin metadevice features	280
20	FTS support across Solutions Enabler features (Enginuity 5876)	309
21	Provisioning limits for arrays running HYPERMAX OS 5977 or higher	312
22	Provisioning limits for VMAX arrays running Enginuity 5874 through 5876	313
23	LUN base/offset scenarios for multiple hosts with skip holes	380
24	Host platforms and interface configuration flags	382
25	Virtual LUN requirements by features	400
26	Action descriptions for symmigrate command	403
27	Audit log session information	414
28	Device types - HYPERMAX OS 5977	450
29	Device types - Enginuity 5876 and lower	451
30	Performance display options	487
31	Change Tracker command summary	503
32	SYMCLI report options	511
33	Optimizer CLI command actions	523
34	Optimizer control parameters	525
35	Fields in Optimizer display	530
36	EMC Double Checksum for Oracle functionality	552
37	symchksum syntax and descriptions	557
38	Log file configuration options	574
39	Control and remote device guidelines	594
40	SYMCLI Open Replicator command summary	596
41	Verifying session state	615
42	Device control operations allowed for ORS control device states with host	664
43	TimeFinder/Mirror source device is control device for rcopy push session	665
44	TimeFinder/Mirror target device is control device for rcopy push session	666
45	TimeFinder/Mirror source device is control device for rcopy pull session	666
46	TimeFinder/Mirror target device is control device for rcopy pull session	667
47	TimeFinder/Clone source device is control device for rcopy push session	667
48	TimeFinder/Clone target device is control device for rcopy push session	668
49	TimeFinder/Clone source device is control device for rcopy pull session	669
50	TimeFinder/Clone target device is control device for rcopy pull session	669

51 TimeFinder/Snap VX source device is control device for rcopy push session..... 670
52 TimeFinder/SnapVX target device is control device for rcopy push session..... 671
53 TimeFinder/Snap VXsource device is control device for rcopy pull session 671
54 TimeFinder/SnapVXtarget device is control device for rcopy pull session..... 672
55 SRDF R1 mirror is control device for rcopy push session 673
56 SRDF R2 mirror is control device for rcopy push session 673
57 SRDF R1 mirror is control device for rcopy pull session 674
58 SRDF R2 mirror is control device for rcopy pull session 675
59 ORS operations allowed by device type 676

PREFACE

As part of an effort to improve its product lines, EMC periodically releases revisions of its software and hardware. Therefore, some functions described in this document might not be supported by all versions of the software or hardware currently in use. The product release notes provide the most up-to-date information on product features.

Contact your EMC representative if a product does not function properly or does not function as described in this document.

Note: This document was accurate at publication time. New versions of this document might be released on the EMC online support website. Check the EMC online support website to ensure that you are using the latest version of this document.

Purpose

This document describes how to use EMC® Solutions Enabler SYMCLI commands to configure and manage VMAX® arrays.

- Part 1, "Setting up and Configuring Storage Arrays," describes how to use the SYMCLI configuration commands to set up your array devices.
- Part 2, "Querying and Reporting," describes how to use the SYMCLI query commands and various reporting tools to monitor and manage your storage array performance.
- Part 3, "Migrating Data," describes how to use SYMCLI Open Replicator to copy data between arrays and SYMCLI Federated Live Migration to non-disruptively migrate data from DMX arrays to VMAX arrays.

Audience

This document is part of the Solutions Enabler documentation set, and is intended for use by advanced command-line users and script programmers to manage various types of control operations on VMAX arrays and devices using the Solutions Enabler SYMCLI commands.

Related documentation

The following documents provide additional information about Solutions Enabler:

- *EMC Solutions Enabler V8.0.1 Release Notes* — Describe new features and any known limitations.
- *EMC Solutions Enabler Installation Guide* — Provides host-specific installation instructions.
- *EMC Solutions Enabler CLI Command Reference* — Documents the SYMCLI commands, daemons, error codes and option file parameters provided with the Solutions Enabler man pages.
- *EMC Solutions Enabler SRDF Family CLI User Guide* — Describes how to configure and manage SRDF environments using SYMCLI commands.

- *EMC Solutions Enabler TimeFinder Family CLI User Guide* — Describes how to configure and manage TimeFinder environments using SYMCLI commands.
- *EMC Solutions Enabler SRM CLI User Guide* — Provides Storage Resource Management (SRM) information related to various data objects and data handling facilities.
- *EMC VMAX Family Security Configuration Guide* — Describes how to configure VMAX Family security settings.

The following documents provide additional information:

- *EMC VMAX3 Family Documentation Set* — Contains documentation related to the VMAX 100K, 200K, and 400K arrays.
- *EMC VMAX Family (10K, 20K, 40K) Documentation Set* — Contains documentation related to the VMAX 10K, 20K, and 40K arrays.
- *EMC VMAX3 Family with HYPERMAX OS Release Notes* — Describe new features and any known limitations.
- *EMC VMAX Family Viewer for Desktop and iPad®* — Illustrates system hardware, incrementally scalable system configurations, and available host connectivity offered for VMAX arrays.
- E-Lab™ Interoperability Navigator (ELN) — Provides a web-based interoperability and solution search portal. You can find the ELN at https://elabnavigator.EMC.com.
- SolVe Desktop — Provides links to documentation, procedures for common tasks, and connectivity information for 2-site and 3-site SRDF configurations. To download the SolVe Desktop tool, go to EMC Online Support at https://support.EMC.com and search for SolVe Desktop. Download the SolVe Desktop and load the VMAX Family and DMX procedure generator.

  Note: You need to authenticate (authorize) your SolVe Desktop. Once it is installed, please familiarize yourself with the information under Help tab.

Conventions used in this document

EMC uses the following conventions for special notices:

⚠CAUTION

CAUTION, used with the safety alert symbol, indicates a hazardous situation which, if not avoided, could result in minor or moderate injury.

Note: A note presents information that is important, but not hazard-related.

IMPORTANT

An important notice contains information essential to software or hardware operation.

Typographical conventions

EMC uses the following type style conventions in this document:

Normal	Used in running (nonprocedural) text for: • Names of interface elements, such as names of windows, dialog boxes, buttons, fields, and menus • Names of resources, attributes, pools, Boolean expressions, buttons, DQL statements, keywords, clauses, environment variables, functions, and utilities • URLs, pathnames, filenames, directory names, computer names, links, groups, service keys, file systems, and notifications
Bold	Used in running (nonprocedural) text for names of commands, daemons, options, programs, processes, services, applications, utilities, kernels, notifications, system calls, and man pages Used in procedures for: • Names of interface elements, such as names of windows, dialog boxes, buttons, fields, and menus • What the user specifically selects, clicks, presses, or types
Italic	Used in all text (including procedures) for: • Full titles of publications referenced in text • Emphasis, for example, a new term • Variables
`Courier`	Used for: • System output, such as an error message or script • URLs, complete paths, filenames, prompts, and syntax when shown outside of running text
`Courier bold`	Used for specific user input, such as commands
`Courier italic`	Used in procedures for: • Variables on the command line • User input variables
< >	Angle brackets enclose parameter or variable values supplied by the user
[ ]	Square brackets enclose optional values
|	Vertical bar indicates alternate selections — the bar means “or”
{ }	Braces enclose content that the user must specify, such as x or y or z
...	Ellipses indicate nonessential information omitted from the example

Where to get help

EMC support, product, and licensing information can be obtained on EMC Online Support, as described next.

Note: To open a service request through EMC Online Support, you must have a valid support agreement. Contact your EMC sales representative for details about obtaining a valid support agreement or to answer any questions about your account.

Product information

For documentation, release notes, software updates, or for information about EMC products, licensing, and service, go to EMC Online Support (registration required) at:

https://support.EMC.com

Technical support

EMC offers a variety of support options.

Support by Product — EMC offers consolidated, product-specific information on the Web at:

https://support.EMC.com/products

The Support by Product web pages offer quick links to Documentation, White Papers, Advisories (such as frequently used Knowledgebase articles), and Downloads, as well as more dynamic content, such as presentations, discussion, relevant Customer Support Forum entries, and a link to EMC Live Chat.

EMC Live Chat — Open a Chat or instant message session with an EMC Support Engineer.

eLicensing support

To activate your entitlements and obtain your VMAX license files, visit the Service Center on https://support.EMC.com, as directed on your License Authorization Code (LAC) letter emailed to you.

For help with missing or incorrect entitlements after activation (that is, expected functionality remains unavailable because it is not licensed), contact your EMC Account Representative or Authorized Reseller.

For help with any errors applying license files through Solutions Enabler, contact the EMC Customer Support Center.

If you are missing a LAC letter, or require further instructions on activating your licenses through the Online Support site, contact EMC's worldwide Licensing team at licensing@emc.com or call:

- North America, Latin America, APJK, Australia, New Zealand: SVC4EMC (800-782-4362) and follow the voice prompts.
- EMEA: +353 (0) 21 4879862 and follow the voice prompts.

Your comments

Your suggestions will help us continue to improve the accuracy, organization, and overall quality of the user publications. Send your opinions of this document to:

VMAXContentFeedback@emc.com

PART 1

Setting up and Configuring Storage Arrays

Part 1 of this guide describes how to set up and configure storage array devices using the Solutions Enabler SYMCLI commands.

Chapter 1, "Introduction," describes the Solutions Enabler Symmetrix command line interface (SYMCLI) terminology and the interface between the host and the storage arrays.

Chapter 2, "Discovery,"describes the SYMCLI discovery process and how to direct requests to retrieve array-level or device-level information from it using SYMCLI commands.

Chapter 3, "User and Group Authorization," describes how to set up authorization using the SYMCLI symauth command.

Chapter 4, "Host-based Access Control," describes how to set up and maintain an access controlled environment using the SYMCLI symacl command.

Chapter 5, "Grouping Devices," describes the benefits of grouping storage array devices, group types and how to create and modify them, and how to use the GNS (Group Name Service) repository.

Chapter 6, "Fully Automated Storage Tiering," describes Fully Automated Storage Tiering (FAST) operations.

Chapter 7, "Managing Configuration Changes," describes how to perform configuration change operations using the SYMCLI symconfigure command.

Chapter 8, "Configuring Virtual Provisioning," describes Virtual Provisioning™ concepts and how to perform provisioning operations using the SYMCLI.

Chapter 9, "Federated Tiered Storage," describes Federated Tiered Storage (FTS) concepts and explains how to configure and manage FTS using the SYMCLI.

Chapter 10, "Masking Devices with Auto-Provisioning Groups," explains how to confine host access to array devices using Auto-provisioning Groups and the SYMCLI symaccess command.

Chapter 11, "Masking Devices (Enginuity 5773 or lower)," describes how to confine host access to array devices using SYMCLI device masking commands for DMX arrays running Enginuity 5773 or lower.

Chapter 12, "Managing Quality of Service," describes how to manage the Quality of Service (QoS) on devices in your storage environment and how to manage dynamic cache partitions in your array using the SYMCLI symqos command.

Chapter 13, "Virtual LUN Technology (Enginuity 5874 through 5876)," explains how to change device protection or improve array performance using Enhanced Virtual LUN Technology and Virtual LUN VP mobility.

CHAPTER 1
Introduction

This chapter introduces the Solutions Enabler Symmetrix command line interface (SYMCLI), which is used to manage your storage environment.

- Overview 28
- Getting SYMCLI help 29
- Command input tips 29
- Setting compatibility 31

Overview

EMC® Solutions Enabler software provides your host with the Symmetrix® Command Line Interface (SYMCLI). The SYMCLI is a comprehensive command set for managing your storage environment. SYMCLI commands can be invoked on the command line or within scripts. These commands can be used to monitor device configuration and status, and perform control operations on devices and data objects within your storage environment.

With SYMCLI you can:

- Discover the storage environment
- Configure user authorization
- Create device and storage groups
- Configure FAST (Enginuity 5874 through 5876)
- Provision storage
- Create e-disks
- Query configuration data for SCSI, array, device, disk, and virtual environments
- View performance statistics
- View log files
- Migrate data (Federated Live Migration - Enginuity 5773 through 5876)
- Perform TimeFinder and SRDF control operations

Note: TimeFinder and SRDF CLI operational commands are not discussed in this guide. Refer to the *Solutions Enabler TimeFinder CLI Guide* and the *Solutions Enabler SRDF CLI Guide* for SYMCLI commands for these operations.

The SYMCLI resides on a host system to monitor and perform control operations on DMX and VMAX Family storage arrays. SYMCLI commands are invoked from the host operating system command line (shell). The SYMCLI commands are built on top of SYMAPI library functions, which use system calls that generate low-level I/O SCSI commands to the storage arrays. To reduce the number of inquiries from the host to the storage arrays, configuration and status information is maintained in a host database file (called the Symmetrix configuration database; `symapi_db.bin` by default).

Getting SYMCLI help

This section provides some tips on obtaining help for SYMCLI commands.

Top level command help

You can use the `symcli` command to obtain information about all the commands included in the SYMCLI. Table 1 describes the various `symcli` help options.

Table 1 SYMCLI help options

Command	Description
`symcli`	Provides the version number of the installed command line interface.
`symcli -h`	Describes how to use the SYMCLI command.
`symcli -v`	Displays the SYMCLI commands and a brief description of each command.
`symcli -env`	Displays a list of the environment variables that can be set for a SYMCLI session.
`symcli -def`	Displays a list of the environment variables that are set for the current SYMCLI session.

Individual command help

Type *`command`* `-h` to display command line help for a specified command.

For example, to display help for the `symcfg` command, enter:

```
symcfg - h
```

In addition, each command has its own viewable man page for quick online reference purposes. You can view these man pages in any text editor.

For example on UNIX, to view the man page for the `symcfg` command, enter:

```
man symcfg
```

- In a UNIX environment, you need to have the SYMCLI man page directory (/usr/symcli/man/) included in your MANPATH environment variable.
- On Windows, the default directory for man pages is C:\Program Files\EMC\symcli\man

Command input tips

This section provides some tips on streamlining your command line entries.

Time-saving tips

The following are some time-saving command input tips:

- You can abbreviate actions by typing only three or more unique characters as follows:

  ```
  symcfg discover
  ```

  can be:

  ```
  symcfg dis
  ```

- You can specify just the last two or more digits of the storage array identifier (SID) in place of the entire 12-digit SID as follows:

  ```
  symcfg -sid 000002304324 sync
  ```

  can be:

  ```
  symcfg -sid 24 sync
  ```

- Actions, keywords, and options are case-insensitive as follows:

  ```
  symcfg LIST
  ```

  or:

  ```
  symcfg list
  ```

You can abbreviate physical device names as follows:

```
sympd show /dev/rdsk/c2t1d1s2
```

can be:

```
sympd show c2t1d1s2
```

Viewing truncated output

In some cases, the output data from SYMCLI list commands may exceed the available column width. In these cases the returned data is truncated and appended with an asterisk (*). To view the full data, use the `show` command or verbose option (`-v`), when available.

Referencing devices and objects

You can use a number of different terms to identify and specify a storage array device with the SYMCLI commands:

PdevName or `pd` — Indicates a physical (host) device name

SymDevName or `dev` — Indicates a VMAX or DMX array device name

LdevName or `ld` — Indicates a logical device name

`-devs` — Indicates a range or list of devices

Environment variables

You can preset environment variables to streamline your command line session. By setting environment variables to commonly used values, you eliminate the need to specify those arguments on the command line.

Displaying environment variables

You can issue `symcli -env` to view a list of environment variables that can be set for your SYMCLI session and `symcli -def` to view the environment variables you currently have set.

Enabling

To set an environment variable, use the form `setenv variable_name value`. For example, you can enable the SYMCLI_VERBOSE variable to set the default command behavior to always display command output with details:

```
setenv SYMCLI_VERBOSE 1
```

Disabling

To turn off an environment variable, use the command `unsetenv`. You can turn off or remove any environment variable. For example, to turn off the verbose mode, enter:

```
unsetenv SYMCLI_VERBOSE
```

Presetting names and IDs

To reduce repeated key strokes for a group of commands that require the same argument, you can preset the device group name or the VMAX array ID to a specified default value. You can preset the following variables to a specified value for the same argument of a command set:

`SYMCLI_DG` — Defines the device group name for all `-g` arguments.

`SYMCLI_SID` — Defines the array ID for all `-sid` arguments.

For example, to set the array ID to `100200000567` for the next group of commands in a sequence, enter:

```
setenv SYMCLI_SID 100200000567
```

Setting group names to display full names

By default, if group names are longer than the allowed space allotment, the displayed name is truncated in the output for the `symcg`, `symdg`, and `symsg list` commands.

To allow the full name to be displayed in the list command output, set the CLI environment variable to SYMCLI_FULL_NAME. For example, to set the array ID to `100200000567` to display full group names for the `symcg`, `symdg`, and `symsg list` commands, enter:

```
setenv SYMCLI_FULL_NAME 100200000567
```

To turn off the full name environment variable, use the command `unsetenv`.

Setting compatibility

This section describes Solutions Enabler compatibility information and restrictions.

Prior Solutions Enabler scripts

If you developed specialized scripts using a prior version of Solutions Enabler and are concerned that they will not work because of output changes in the latest release, you can run your commands or scripts in compatibility mode. Compatibility mode specifies the command output reporting style to be compatible with prior SYMCLI versions.

Note: While running older scripts in compatibility mode, you may not return output that is only available in later releases.

You can set your compatibility mode as a global environment variable using `setenv SYMCLI_MODE` or at the command line using the `-mode` option on any command. This specifies the command output reporting style to be compatible with prior SYMCLI versions. Possible values are V74, V75, and V76.

For example, to set the compatibility mode to Solutions Enabler V7.4, enter the following:

```
symcfg list -mode V74
```

Attempts to specify a compatibility mode earlier than V7.4 fails with an Invalid mode error. For example:

```
symauth -sid 111 -mode V72
'V72': Invalid mode specified.
```

Compatibility for CLI displays

Starting with Solutions Enabler V8.0.1, the number of logical devices that can be created on a single array is 128K. To support this increase, a number of CLI output displays have been adjusted to permit larger string representations of VMAX device numbers as hexadecimal number strings.

Unless noted otherwise, you can issue output CLI display commands with the `-mode V76` option to display output consistent with Solutions Enabler V7.6. If 5-character devices are included in these displays, the lines containing such devices shift any additional characters on the line to the right, while column headings remain unchanged. This may result in misaligned displays when using `-mode V76`.

Client/server compatibility restrictions

In Solutions Enabler client/server mode, client host operations are automatically forwarded to the `storsrvd` daemon on a server host for execution.

Solutions Enabler V8.0.1 has additional restrictions limiting the interoperability between client and server with different versions of Solutions Enabler. The following restrictions are enforced in this release:

- SYMAPI client connections to servers running Solutions Enabler versions lower than V8.0.1 are rejected.
- SYMAPI server connections from clients running Solutions Enabler earlier than V7.4 are rejected with the error code: `SYMAPI_C_NET_VERSION_TOO_OLD`

  The remote connection is refused, as the client SYMAPI version is not supported by the server.
- SYMAPI server connections from clients with a Solutions Enabler version later than server are rejected with error code: `SYMAPI_C_NET_VERSION_TOO_NEW`

  The remote connection is refused, as the client cannot have a newer SYMAPI version than the server.

- When comparing client and server versions, the major version, minor version, edit level, and patch level are compared.
- When comparing database versions, only the major and minor versions are compared (for example: 7.4, 7.5). The edit and patch levels are ignored.
- Attempts to open a database file written by a version prior to V7.4 fail with the following error code:

  ```
  SYMAPI_C_DB_VERSION_TOO_OLD
  The SYMAPI database file cannot be used because it was written by a
  version of SYMAPI that is no longer supported.
  ```

CHAPTER 2
Discovery

This chapter describes the SYMCLI discovery process.

- Overview 36
- Verifying host configuration database is synchronized 36
- Scanning for new devices 37
- Authorizing connectivity 37
- Configuring virtual disk mapping 38
- Working with the host configuration database file 39
- Synchronizing configuration data 41

Overview

Before using the SYMCLI commands, you need to run the `symcfg discover` command to build your configuration (SYMAPI) database. This needs to be done once after installation, and after any changes are made to your array configuration.

Discovery refers to the process by which array, volume-level configuration and status information is retrieved. Discovered configuration and status data for all arrays, as well as their directors and volumes, is maintained in a configuration database file on each host. Once you have discovered your environment, you can direct information requests to retrieve array-level (high-level) data or device-level (low-level) information from it using SYMCLI commands.

During your first command line session, or if a configuration change has occurred, the configuration database must be built or rebuilt with the most complete and current information for all physical devices connected to your host. The discovery syntax options are:

```
symcfg

   discover [-all | -pdev [-sid SymmID] | -sid SymmID]
            [-cache | -nocache]
```

To scan the hardware and rebuild the database, enter:

```
symcfg discover
```

This command scans all SCSI buses, collects information about all the arrays and devices found, and rebuilds the database with the collected device information and parameters from all local and remotely attached devices.

You can limit the discovery to only physical device information by specifying the `-pdev` option, or a specific array by specifying the `-sid` option (you can use this option with the `-pdev` option to return only physical device information for a specific array). For more information about the configuration database, refer to "Working with the host configuration database file" on page 39.

Verifying host configuration database is synchronized

Use the `symcfg verify` command to determine if the host configuration database file is synchronized with the current configuration of a specific array. If they are out of sync, the verify action returns code 24 (`CLI_C_NOT_IN_SYNC`). If they are in sync, the verify action returns code 0 (the `CLI_C_SUCCESS` value) as shown below using the `echo $status` command.

```
symcfg verify -sid 814

The Symmetrix configuration and the database file are in sync.

echo $status
0
```

You can issue the `symcfg list -status` command to list all arrays connected to your host, and show whether the configuration has changed and if the array was discovered during the last discovery operation. The following is an example output:

```
symcfg list -status

                          S Y M M E T R I X

                                      Mcode
SymmID        Attachment  Model       Version  Config Changed  Discovered

000192600305  Local       VMAX        5977     Yes             Yes
000192600321  Local       VMAX        5977     Yes             Yes
000192600198  Remote      VMAX        5977     Yes             Yes
000192600256  Remote      VMAX        5977     Yes             Yes
000192600282  Remote      VMAX        5977     No              Yes
000192600284  Remote      VMAX        5977     No              Yes
```

Scanning for new devices

When new devices are mapped to a host on nodes that have not yet been created, those nodes must exist before the discovery operation can find the devices along their path. A `symcfg scan` command searches your environment for devices accessible to a given host, creates the nodes for the new or changed devices, and activates the necessary processing on the host system to recreate a list of accessible devices. It should be followed by a `symcfg discover -symmetrix`, if the changes are associated with array devices.

Authorizing connectivity

Some arrays may require authorization information to provide access to the array. Issue the `symcfg authorization` command to supply this information for use in subsequent discovery operations. The `symcfg authorization` command allows you to list, add, update, or delete this connectivity information. The `update` action allows you to update the password of an existing entry.

Use the following syntax:

```
symcfg

       authorization list [-hyperv] [-v]

        authorization add | update -host HostName
           -username UserName -password PassWord
           [-namespace NameSpace] [-ahost HostName]
           [<-vmport | -port> port] [-vmware]

        authorization add | update -host HostName
           -username UserName -password PassWord
           [-hyperv]

        authorization delete -host HostName
            -username UserName
           [-namespace NameSpace] [-vmware]
```

Configuring virtual disk mapping

Solutions Enabler can resolve all virtual disks that have only one array device in the VMware datastore (supported with ESX 3.5 and higher). To use VMware virtual disk mapping, you must configure credentials using the following `symcfg auth` command. For example, enter:

```
symcfg auth add -host HostName -username root -password PassWord
-namespace -vmware
```

In addition, copy the SSL certificate from the server (for example, `/etc/vmware/ssl/rui.crt`) to the VM OS as `/var/symapi/config/viclient_cert.pem`.

When virtual disk mapping is configured, the `syminq` command returns the virtual disk information:

./syminq

```
      Device              Product                    Device
---------------- --------------------------- ---------------------
Name      Type   Vendor    ID           Rev  Ser Num      Cap (KB)
---------------- --------------------------- ---------------------
/dev/sda         VMware    Virtual disk 1.0  N/A          20971520
/dev/sdc         VMware    Virtual disk 1.0  N/A           8388608
/dev/sdd         EMC       SYMMETRIX    5773 8600022000      92160
/dev/sde         VMware    Virtual disk 1.0  N/A           8388608
/dev/dm-0        VMware    Virtual disk 1.0  N/A           8388608
```

Note: An expected capacity difference is visible between the output of `syminq` and `sympd list`. When creating the virtual disk, it can use all the space or less space than the array device; so the `syminq` output may show smaller capacity than what `sympd` shows. In some cases, `syminq` shows larger capacity than `sympd`, because the VMware datastore (or virtual disk) can have multiple array devices.

The `sympd` command displays virtual disks as array devices. The example below shows an output of `sympd list`:

```
sympd list -sid 266

Symmetrix ID: 000194900266

        Device Name           Dir                 Device
---------------------------- ----- -------------------------------------
                                                                    Cap
Physical                Sym   SA :P Config        Attribute    Sts   (MB)
---------------------------- ----- -------------------------------------
/dev/sdl                00044 01E:0 2-Way Mir     N/Grp'd ACLX RW       6
/dev/sdas               00168 07E:0 2-Way Mir     N/Grp'd      RW      28
/dev/sds                006DC 07E:0 TDEV          N/Grp'd      NR    1934
```

Working with the host configuration database file

The host configuration database (`.bin`) file, which is stored on the server host system, contains the physical configuration information of SCSI devices and array parameters that define your entire storage complex. More than one configuration database file may be required to support your operational needs.

The host configuration database is sometimes referred to SYMAPI database (because of how the file is named), or the Symmetrix database file. All of these names are referring to the same configuration database file, `symapi_db.bin`, described next.

Database file location

On UNIX, the default pathname for the configuration database file is:

```
/var/symapi/db/symapi_db.bin
```

On Windows, the default configuration database path is:

```
C:\Program Files\EMC\SYMAPI\db\symapi_db.bin
```

On OpenVMS, the default configuration database path is:

```
SYMAPI$DB:symapi_db.bin
```

You can create an additional configuration database (`.bin`) file to meet your requirements.

Note: In a multi-database environment, verify that you are operating on the correct database. Always confirm the current database before applying a command. For safe command line environments, it is recommended that you use the common (default) database.

Database file lock

Solutions Enabler utilizes a configuration database locking file. The lock file is created automatically and is given the same name as the configuration database file, but it is appended with an *_xlock* suffix. For example, `symapi_db.bin_xlock`.

Solutions Enabler uses this lock file to serialize access to the configuration database file. It contains no data and is only used as a lock.

If you protect the `symapi_db.bin` file to restrict the users permitted to perform Solutions Enabler management operations, this lock file should be protected as well. Both `symapi_db.bin` and `symapi_db.bin_xlock` should be given the same level of protection.

Changing the current database file

To check which host configuration database file is being used, enter:

```
symcli -def
```

To change the current host configuration database file, you need to modify the environment variable SYMCLI_DB_FILE.

For example, to change the database file to `symbackup_db.bin` from a UNIX host in C shell (csh), enter:

```
setenv SYMCLI_DB_FILE /var/symapi/db/symbackup_db.bin
```

To perform the same operation on Windows, enter:

```
set SYMCLI_DB_FILE=C:\Program Files\EMC\Symapi\db\symbackup_db.bin
```

Changing the database in client/server mode

For security reasons, in client/server mode the configuration database file must be located within the default database directory on the server host.

On UNIX, the default pathname for the configuration database file is:

```
/var/symapi/db
```

On Windows, the default configuration database path is:

```
C:\Program Files\EMC\Symapi\db
```

Database access modes

The SYMCLI commands utilize different modes to access the host configuration database file:

read/write — Commands that control and/or modify database parameters, read the database file into memory, and provide simultaneous modification of both the in-memory database and the database file. During this access cycle, the database file is locked.

read/no write — Commands that list or show database parameters, read the database file into memory and allow modifications to the in-memory database. No modifications to the database file occur. During the access cycle, the database file is not locked.

Command modes: Online and offline

You can run SYMCLI commands in online or offline mode. Commands that execute in online mode, such as configuration control operations, automatically attempt to gather the latest state and mode information from the arrays, and update the in-memory database and configuration database file on the host. If a configuration change has occurred, commands that execute in online mode will attempt to discover the changed entity to retrieve and load any updated configuration information.

Commands that can execute in offline mode, such as `symcfg list`, retrieve data exclusively from the configuration database.

Database configuration information

The `symcfg -db` command returns basic configuration information about the current database in use. The following example may not match the values present in your configuration:

```
symcfg -db

  Type of SYMAPI Database                          : Full
  Host Node Name which discovered the DB           : api99
  Host OS Type which discovered the DB             : AIX

  Version of SYMAPI Library which discovered the DB : 7.2.X.X (Edit Level: 1102)
  Version of SYMAPI Library which wrote the DB      : 7.2.X.X (Edit Level: 1102)
  Min Edit Level of SYMAPI Lib Reqd to Read the DB  : 1101

  Time Database Was Last Synchronized              : Thu Oct 21 09:04:06 2010
  Time Any Device Group Was Last Modified          : Thu Oct 21 13:52:03 2010

  GNS State                                        : Enabled
```

Note: The above output includes the minimum edit level of the SYMAPI library required to read the database. Any SYMAPI library with this version or higher and edit level or higher can read the current database.

Synchronizing configuration data

Once a discovery operation has occurred, you can synchronize configuration data in the host configuration database file with the storage environment, which is less performance-intensive than a full discover operation.

A *sync* operation interrogates just the known arrays (previously discovered) that are accessible from the host, and updates the configuration and status information in the configuration database file. In addition, this synchronization can be limited to a specific array or array type, where a configuration change may have occurred.

Sync operations

A sync operation only updates state, mode, and configuration information (if a configuration change has occurred). It does not update statistical information. Chapter 15, "Statistics," provides information about obtaining array performance statistics.

The `sync` option updates data for all known entities in the configuration database file, but does not scan for SCSI devices on the host. That means that any newly configured physical devices will not be added to the database during a sync operation.

Synchronizing data is an array-specific operation. The `symcfg sync` command performs a basic sync of all known array entities. The following options limit the scope to a specific data set:

- `[-sid SymmID]` — Limits the scope to a specific array
- `[-rdf|` — SRDF information
- `-bcv|` — BCV information
- `-local|` — Local array information

- `-dirsts|` — Director status information
- `-snap|` — TimeFinder/Snap information
- `-cfgmgr|` — Disk space and array configuration metrics
- `-rcopy|` — R-Copy information
- `-env_data|` — Environmental data
- `-fast|` — Fully-Automated Storage Tiering (FAST) information
- `-tier|` — Array tier definition information
- `-sg|` — Storage group information
- `-vpdata]` — Virtual Provisioning data

Inhibiting database synchronization

You can force some commands to operate in offline mode using the environment variable SYMCLI_OFFLINE, which inhibits accessing the array to update the database.

For example, to globally force these commands to their offline mode (`-offline` option) from a UNIX host in C shell (csh), enter:

```
setenv SYMCLI_OFFLINE 1
```

You may need to refresh the database if you execute commands while offline mode is enabled or execute commands that normally run in the offline mode, such as `symcfg list`. Most display commands can be run in the offline mode, which provides the fastest response, and does not require access to the array.

CHAPTER 3
User and Group Authorization

This chapter describes how to set up user and group authorization using the SYMCLI `symauth` command.

- Overview 44
- Managing authorization settings 47
- Monitoring authorization 52

Overview

User authorization is a tool for restricting the management operations individual users or groups can perform on an array. User authorization and the host-based Access Control database are independent utilities but can be used together for maximum security. Refer to "Host-based Access Control" on page 55 for more information on the Access Control database.

You can use SYMCLI `symauth` command to map a user or group to a specific role that defines the operations the user or group can perform on either an entire array or on individual components within an array. Authorization is configured independently for each array.

For more information about the `symauth` command syntax, refer to the *EMC Solutions Enabler SYMCLI Command Reference Guide*.

User roles

A role is a predefined set of permissions that determine the operations a user can perform. Unlike host-based access control, a user is assigned a particular role for the entire array or certain components within the array. For each array, a given user or group can be assigned up to four roles. The roles themselves cannot be modified.

To list the roles supported by an array, enter:

```
symauth list -roles
```

The following roles are defined in Solutions Enabler:

- **None** — Has no rights.
- **Monitor** — Performs read-only operations on an array excluding the ability to read the audit log or Access Control definitions.
- **PerfMonitor** —Includes Monitor role permissions and grants additional privileges within the performance component of Unisphere for VMAX application to set up various alerts and update thresholds to monitor array performance.
- **StorageAdmin** — Performs all management operations on an array or on individual components within an array in addition to all monitor operations. This is the only role that can be given access rights to specific components within an array and is limited to Virtualization domain users. Refer to "Authentication types" on page 46 for user ID format to assign Storage Admin role for virtualization domain user.

  A StorageAdmin can modify the GNS group definitions and monitor all operations (even if only granted rights to one component); includes application performance monitor privileges. For a storage group or a thin pool, the StorageAdmin can perform the following operations:

  - Storage groups
    - Add devices to a storage group
    - Delete devices from a storage group
  - Thin pools (Enginuity 5876 or lower)
    - Create TDEVs (permitted if user has rights to any thin pool on the array)
    - Delete unbound TDEVs (permitted if user has rights to any thin pool on the array)
    - Bind TDEVs to a thin pool
    - Unbind TDEVs from a thin pool
    - Make TDEVs bound to a thin pool ready or not ready

- **SecurityAdmin** — Performs security operations (`symaudit`, `symacl`, `symauth`) on an array in addition to all monitor operations. Users or groups assigned the SecurityAdmin or Admin roles can create or delete component-specific authorization rules. The SecurityAdmin also has all Auditor rights.
- **Admin** — Performs all operations on an array, including security operations and monitor operations. The Admin also has StorageAdmin rights, SecurityAdmin rights, and application performance monitoring privileges.
- **Auditor** — Grants the ability to view, but not modify, security settings for an array (including reading the audit log, `symacl list`, and `symauth`) in addition to all monitor operations. This is the minimum role required to view the array audit log.

Multiple user roles

Starting with V8.0.1, Solutions Enabler allows a user or group to be assigned to as many as four authorization roles. The `symauth` command has been enhanced to enable you to assign multiple roles separated with a '+' character. For example:

```
StorageAdmin+Auditor+Monitor
```

Note: If any other role is combined with a role of None, the None role is ignored.

A new `remove` command allows you to remove roles for a user. The remove operation only works after the user has been assigned multiple roles. If a user has only one role, `remove` cannot be used to remove that role. For example:

```
symauth -sid <SymmID> commit <<!

assign user  H:mars\smith   to role StorageAdmin+Auditor;
assign group D:Eng\Sup      to role SecurityAdmin+PerfMonitor;
assign user  H:mars\smith   to role Monitor;
remove group D:Eng\Sup      from role PerfMonitor;
remove user  H:mars\smith   from role Auditor;
```

Note: Solutions Enabler versions 7.6 or earlier support only one role per user or group. When a user has multiple roles defined on an array running Solutions Enabler 8.0.1, older versions of Solutions Enabler will show only the highest assigned role for that user.

Authorization rules

When a user or group attempts access to an array or one of its components, Solutions Enabler performs the following authorization checks:

a. Verifies if user has access to the entire array.

b. Checks if group has access to the entire array.

c. Verifies if user has access to the ThinPool or StorageGroup component within the array.

d. Checks if group has access to the ThinPool or StorageGroup component within the array.

When determining the rights of a user or group, each of the above authorization checks is performed. The rights granted by each check are combined together to form an overall set of rights for the user or group. For example:

- User Larry is assigned the SecurityAdmin role.
- Group Finance is assigned the Monitor role.

Because User Larry belongs to Group Finance, he is granted both SecurityAdmin and Monitor rights.

User ID formats

Users are identified by the user ID format, *Type:Qualilfier\UserName or GroupName*.

Where:

- *Type* specifies the type of security authority used to authenticate the user. Authentication types are listed below.
- *Qualifier* specifies the domain or host that the user is logged into.
- *Name* specifies the username or groupname relative to that authority. It cannot be greater than 32 characters, and spaces are allowed if delimited with quotes.

Authentication types

The following authentication types are supported:

- `L` — Indicates a user or group authenticated by LDAP. In this case, Domain specifies the domain controller on the LDAP server. For example:

  `L:danube.com\Finance` — Indicates that user group Finance logged in through the domain controller danube.com.
- `C` — Indicates a user or group authenticated by the SMC server. For example:

  `C:Boston\Legal`— Indicates that user group Legal logged in through SMC sever Boston.
- `D` — Indicates a user authenticated by a Windows domain. The qualifier specifies the domain or realm name. For example:

  `D:sales\putman` — User putman logged in through the Windows domain sales.

  `D:jupiter\Sales` — Group Sales logged in through the Windows domain jupiter.
- `H` — Indicates a user authenticated (by logging in) to some host. On Windows, this corresponds to logging into a local account on the host. The qualifier specifies the hostname. For example:

  `H:jupiter\mason` — User mason logged in on host jupiter.

  `H:jupiter\Sales` — Group Sales logged in on host jupiter.
- `V` — Indicates a user authenticated by a Virtualization domain. The qualifier specifies the domain host name. For example:

  `V:host932jx\connor` — User connor logged in through a Virtualization domain host932jx.

Only Virtualization domain users or groups are allowed to be assigned a role of StorageAdmin with access to specific array components such storage groups or thin data pools. For example, to assign thin pool access to a Virtualization domain user:

```
symauth -sid 1234 commit <<!
assign group V:Host932jx\joe to role StorageAdmin for component
ThinPool ThinPool:Sales-2;
!
```

UserName and GroupName formats

Within role definitions, user IDs or group IDs can be either fully qualified (as shown previously), partially qualified, or unqualified. When the hostname or domain portion of the *UserName* parameter or *GroupName* parameter is an asterisk, the asterisk is treated as a wildcard meaning any host or domain. Examples of this include: `H:*\user`, `D:*\user`, and `*\user`. In all other cases, the asterisk is treated as a regular character.

For example, the following are valid formats:

- `D:ENG\Sales` — Fully qualified name with a domain and groupname.
- `D:*\jones` — Partially qualified name that matches username `jones` within any domain.
- `H:HOST\Eng` — Fully qualified name with a hostname and groupname.
- `H:*\jones` — Partially qualified name that matches username `jones` within any host.
- `jones` — Unqualified username that matches any `jones` in any domain on any host.

When a wildcard (such as the asterisk) is used, a given user or group may be matched by more than one mapping in the database. When searching for a role, the authorization mechanism uses the closest ID match that it can find.

- If an exact match (for example, `D:sales\putman`) is found, that is used.
- If a partial match (for example, `D:*\putman`) is found, that is used.
- If an unqualified match (for example, `putman`) is found, that is used; otherwise the user is assigned a role of None.

When using the asterisk to allow wildcarding of user or group names, the `symauth` command may fail with an invalid argument error, if it is used to establish a rule (`assign` or `reassign`) with a user or group that is deemed to be invalid through API validity checks.

Managing authorization settings

Use the `symauth` command to manage user and group authorization. It allows you to enable and disable user and group authorization, set enforcement mode, list defined users and groups with their roles, and assign roles to users and groups.

Identifying the current username

To return the current username (and groupname if applicable) enter:

```
symauth show -username
```

```
Your current username is: D:ENG\ljacobs
Your current groupname is: D:ENG\test
```

The `symauth show -username` command displays all groups to which the user belongs.

```
Your current username:D:Corp\Joe
Your current groupname:D:Corp\Finance
      D:Corp\Sales
      H:HostName\PowerUsers
      ...
```

Group names are sorted alphabetically.

Setting up authorization

It is a good practice to define your user and group role mapping before enabling authorization. At a minimum, you must create at least one mapping for yourself to Admin or SecurityAdmin. You can only enable authorization if you (as shown by `symauth show -username`) are already mapped to a role of either Admin or SecurityAdmin.

The following example provides an overview of the recommended steps for setting up user authorization:

1. Create the user-to-role and group-to-role mappings. At a minimum, map users to the Admin and SecurityAdmin roles when you initially configure authorization. These roles provide the ability to perform authorization control operations. Up to four roles can be assigned to a user or group. You must have one of these mappings created before you can successfully enable authorization. This can be done using a command file, or, on a UNIX host, by supplying a redirection of STDIN, as shown next:

   ```
   symauth -sid 1234 commit <<!
   assign user H:jupiter\laura to role SecurityAdmin+PerfMonitor;
   assign user D:Eng\neil to role Monitor;
   assign user D:Eng\dave to role StorageAdmin+Auditor;
   assign user D:Eng\steve to role Admin;
   assign user D:Eng\paul to role PerfMonitor;
   assign group V:Host932jx\joe to role StorageAdmin for component
   ThinPool ThinPool:Sales-2;
   assign group D:Finance to role Monitor;
   !
   ```

   In the previous example, seven role mappings were created, including one Admin role (`D:Eng\steve`), one SecurityAdmin and PerfMonitor roles (`H:jupiter\laura`), one StorageAdmin and Auditor roles (`D:Eng\dave`), one StorageAdmin role (`V:Host932jx\joe`), one Monitor role (`D:Eng\neil`), and one PerfMonitor role (`D:Eng\paul`).

   Note: When using the asterisk to allow wildcarding of user or group names, the `symauth` command may fail with an invalid augument error, if it is used to establish a rule (`assign` or `reassign`) with a user or group that is deemed to be invalid through API validity checks.

2. Once the role mappings are created, you can then enable authorization (if you assigned yourself the Admin or SecurityAdmin role), as shown below:

   ```
   symauth -sid 097 enable
   ```

In the example above, user authorization is enabled on array 097. Specifying an array ID is optional when enabling user and group authorization.

Enabling and disabling authorization

User and group authorization is disabled by default. Any user can make changes to the authorization control data, including creating and removing user and group role mappings. Once authorization is enabled for an array, only users or groups with the Admin and SecurityAdmin role can change authorization control data. In addition, only a Virtualization Domain user assigned the role of StorageAdmin of an entire array can create or delete component-specific authorization rules. Refer to "Authentication types" on page 46 for assigning component specific authorization.

A user or group granted the role of Admin or SecurityAdmin can enable or disable authorization for an entire array. To enable user and group authorization, enter:

```
symauth -sid 1234 enable
```

Setting enforcement

Once user authorization is enabled, failed access attempts can be enforced in one of two ways. An authorization failure can either result in operation failure or simply log a warning. The level of security enforced, can be set using `symauth -sid <SymmID> set enforcement <advise | enforce>`.

- `advise` — This parameter allows the operation to proceed when user or group authorization is denied, but generates a warning message (less secure) to the Solutions Enabler log file. This is an effective mode for validating your user and group role mapping without preventing access to array functionality.
- `enforce` — This parameter, which is the default setting, causes the operation to fail when user or group authorization is denied (more secure), and logs the authorization attempt with its associated User or Group ID to the common audit log, `symaudit` log, and the Solutions Enabler log file.

This value can also be set using a command file as described in, "Command file usage" on page 50.

Modifying role mappings

Users or groups can be assigned a specific role using the `symauth` command by either supplying the role mappings through a command file, or, on a UNIX host, by supplying a redirection of STDIN. Each line of the command file must contain a valid username or groupname with supported roles (refer to "User roles" on page 44 for supported roles), and end with a semicolon. For further details on using the command file, refer to "Command file usage" on page 50.

To modify these mappings using a command file, use the following command syntax:

```
symauth -sid SymmID commit -file PathName
```

Where *PathName* is the fully qualified path of the file containing the user-to-role mappings.

Commit using STDIN

On a UNIX host, users can be assigned a specific role by supplying a redirection of STDIN, as shown in the example below:

```
symauth -sid 1234 commit <<!
assign user H:jupiter\laura to role Monitor;
assign user D:Eng\neil to role Admin;
assign user lauren to role StorageAdmin for StorGrp SG10;
reassign group Sales to role Auditor;
delete group Marketing for ThinPool ThinPool:Sales-1;
reassign user D:Eng\bob to role Monitor;
reassign user V:Host932jx\joe to role StorageAdmin for ThinPool
ThinPool:Sales-4;
!
```

Note: When using the asterisk to allow wildcarding of user or group names, the `symauth` command may fail with an invalid argument error, if it is used to establish a rule (`assign` or `reassign`) with a user or group that is deemed to be invalid through API validity checks.

Command file usage

You can add authorization management operations to a command file so a sequence of operations can be previewed and committed at once. A command file is a simple text file which must be formatted as specified by the operation below:

Adding a user to roles:

```
assign user UserName to role RoleNames;
```

Note: Starting with Solutions Enabler V8.0.1, the assign operation is additive. The operation adds roles to existing roles and no longer replaces existing roles. A user or group can be assigned up to 4 authorization roles.

Removing roles from user:

```
remove user UserName from role RoleNames;
```

Note: The remove operation is new with Solutions Enabler V8.0.1. The operation allows you to remove roles for a user when the user has multiple roles defined.

Reassigning an existing group to roles:

```
reassign group GroupName to role RoleNames;
```

Deleting a user:

```
delete user UserName;
```

Set enforcement:

```
set enforcement [advise | enforce];
```

Where:

- *UserName* or *GroupName* is the name of a user or a group (32-character maximum). Spaces are allowed if the *UserName* or *Group Name* is quote delimited. Examples of a valid *UserName* include: `"D:domain\joe"`, `"H:host\joe"`, `"domain\joe"`, and `"joe"`. In the case of `"domain\joe"`, the *UserName* is interpreted as `D:domain\joe`. In the case of `joe`, this *UserName* is allowed regardless of domain or host.

  Note: When the hostname or domain portion of a *UserName* or a *UserName* is an asterisk, the asterisk is treated as a wildcard meaning any host. Examples of this include: `H:*\user`, `D:*\user`, and `*\group`. In all other cases, the asterisk is treated as a regular character.

- *RoleNames* are the name of roles to be assigned to or removed from a user or group. Current valid roles include: Admin, SecurityAdmin, StorageAdmin, Auditor, PerfMonitor, Monitor, and None. These are case insensitive. To add multiple roles place a "+" between roles (for example, StorageAdmin + Auditor).

Preview and commit actions

To safely apply any of these authorization actions (command file entries) to the array, perform the following progressive `symauth` operations on the command file:

- Preview — `symauth -sid SymmID preview -file PathName`
  After you first create the command file, the preview operation verifies the syntax and correctness of the contents of the entries in the command file.
- Commit — `symauth -sid SymmID commit -file PathName`
  The commit operation performs both the preview and prepare checks and then commits the contents of the command file to the array.

Backing up and restoring the authorization database

The backup operation saves the contents of the user and group authorization database from an array to the specified file. Use the following command syntax to back up your user authorization database, where *SymmID* is the 12-character ID that specifies the array and *BackupFile* is the fully qualified path and filename of the backup that will be created:

```
symauth -sid SymmID backup -f BackupFile
```

The restore operation re-initializes the user and group authorization database on an array from a previously generated backup file and then re-enables user authorization. The specified file must be created by an earlier backup operation from the same or a different array. Use the following command syntax to restore a previously created backup file, where *SymmID* is the 12-character ID that specifies the array to restore the file and *BackupFile* is the fully qualified path and filename of an existing backup file:

```
symauth -sid SymmID commit -restore -f BackupFile [-noprompt]
```

The restored file must assign the current user a role of Admin or SecurityAdmin; if it does not, the final restore step, which re-enables user authorization will fail. If this occurs, assign a role of Admin or SecurityAdmin to the current user and manually enable user authorization. Alternatively, you can have a user with Admin or SecurityAdmin privileges re-enable user authorization on the array.

Monitoring authorization

Monitoring operations allow you to view the current user authorization settings on a specific array.

To list the user authorization policies in effect, enter:

```
symauth -sid nnnn list
```

To list the roles supported by an array, enter:

```
symauth list -roles
```

To list the component types supported by an array, enter:

```
symauth list -components
```

Listing roles, names, and components

The following example shows the mapping of user and group names to the roles and applicable components for array 1234. As shown, the `StorageAdmin` role has authorized access to an array. For example, user `H:mars\Jones` is granted access to `array 1234,` and this authorization rule is active as noted by the period (.) in the `Flags` column.

Note that user `D:Eng\Joe` is granted authorization to performance monitoring on the array, but the Flag displays `N` (NoEffect) because another rule supersedes this one. In this case user `D:Eng\Joe` also belongs to Group `D:Eng\QA` which is assigned the StorageAdmin role for the array and already grants performance monitoring rights to it's group members.

To list the mappings of roles to user and group names and their applicable components for array 1234, enter:

```
symauth -sid 1234 list -users

   S Y M M E T R I X   A U T H O R I Z A T I O N   U S E R S

Symmetrix ID: 000000001234

                                                                        Flags
Role              User/Group name                Component                E
---------------   ----------------------------   ------------------------ -----
SecurityAdmin     User  Smith                    N/A                        .

StorageAdmin      User  H:mars\Jones             N/A                        .
                  User  H:saturn\Sally           N/A                        .
                  Group D:Eng\QA                 N/A                        .

Auditor           User  H:mars\Jones             N/A                        .
                  Group D:Eng\QA                 N/A                        .

PerfMonitor       User  D:Eng\Joe                N/A                        N

Monitor           User  Moe                      N/A                        .
                  Group Sales                    N/A                        .

Legend for Flags:

(E) : N = Rule has no effect since a different rule grants greater rights.
    : . = Rule is active.
```

Using the -v (verbose) option

Use the `-v` option in `symauth list` when the length of the user/group name and component name exceeds the 79-character limit. To request a multi-line display for authorization information for array 1234, enter the following command:

```
symauth -sid 1234 list -users -v

  S Y M M E T R I X   A U T H O R I Z A T I O N   U S E R S

 Symmetrix ID: 000000001234
Role Name                       SecurityAdmin
User Name                       D:ENG\Smith
Component type
Component name
Flags

Role Name                       StorageAdmin+Auditor
User Name                       H:mars\Jones
Component type
Component name
Flags

Role Name                       StorageAdmin
User Name                       H:saturn\Sally
Component type
Component name
Flags
            ...
```

The `symauth list -users` command supports the offline mode of operation (`-offline`). The list operations retrieves up-to-date information from the specified array. If up-to-date information cannot be retrieved, the operation fails.

With the `-offline` option, Solutions Enabler retrieves information from cached authorization data on disk and does not communicate with the specified array. If there is no cached data on disk, no data displays (as if there is no authorization data on the array).

The following syntax is used:

```
symauth [-sid SymmID] [-offline]

    list

    list -users [-by_domain | -by_role | -by_user]
```

CHAPTER 4
Host-based Access Control

This chapter describes how to set up and perform Access Control actions using the `symacl` command.

- Overview 56
- Creating and modifying access control data 58
- Creating and managing access groups 60
- Creating and managing limited access to access pools 63
- Creating and managing access control entries 65
- Obtaining access control information 69
- Releasing pending locked sessions 70
- Understanding access strategies 71
- Backing up and restoring the access control database 75

Overview

Anyone with access to array-based management software can execute any function on any array device. Many product applications, such as Unisphere for VMAX or SRDF can issue management commands to any device in an array. Open systems hosts can manipulate mainframe devices, Windows hosts can manipulate UNIX data, and vice versa.

The `symacl` access control command allows you to set up and maintain an access control over the array resource (access pools). The command sets up access control mechanisms and changes access control entries through the Access Control database.

For more information about the `symacl` command syntax, refer to the *EMC Solutions Enabler SYMCLI Command Reference Guide*.

Access Control database

An array-based access control database contains all the mechanisms or information to govern access to array access pools.

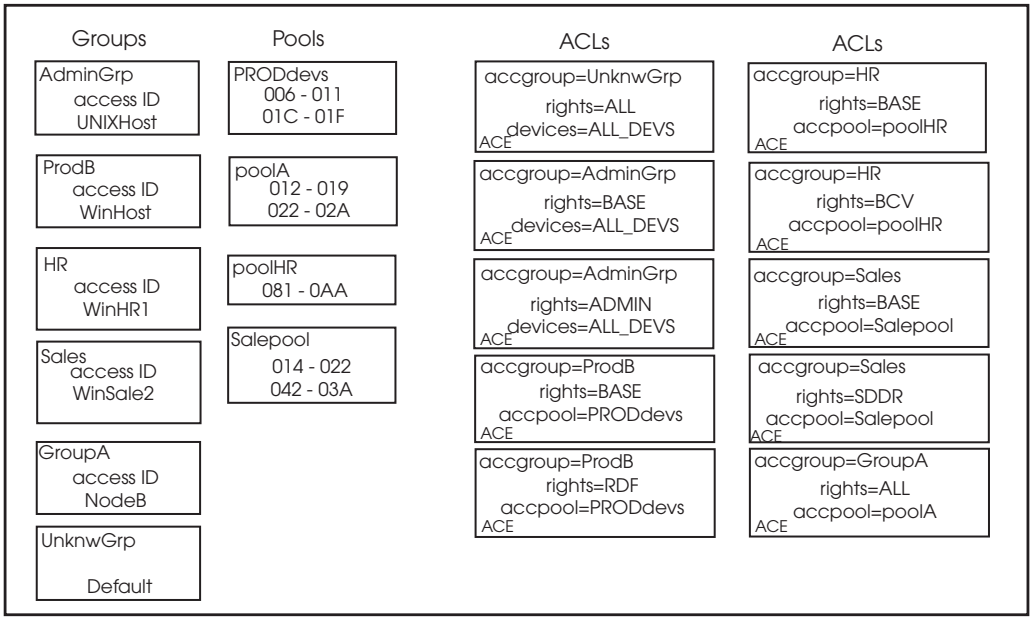


Figure 1 Access Control Lists and Access Control Entries in the Access Control database

The following access control mechanisms comprise the access control database:

- **Access Control groups** — Unique *access IDs* and *names* are assigned (together) to hosts and then sorted into access control *groups* according to similar needs (determined by an Administrator). Access groups are allowed to act on access pools based on *permissions* (*access types*) granted by the Administrator. The unique host ID for open systems can be viewed by running `symacl -unique`.
- **Access pools** — Permissions (or access types), such as BCV, SRDF, ADMIN, are assigned to allow a host to perform certain Solutions Enabler functionality on a specified set of devices. These sets of devices are referred to as access pools or `accpool`. Refer to Table 4 on page 65 for details on access types.

- **Access Control Entry (ACE)** — Once the group and access pool mechanisms are established, the ACEs are created, which grant permissions to these pools. The ACEs of the various access groups (along with groups and pools) are managed and stored in the Access Control database.
- **Access Control List (ACL)** — A group of ACEs that are associated with the same Access Control group.

Securely identifying hosts with access IDs

Access control identifies individual management hosts with an access ID, and there are two different types of access IDs, hardware-based and alternate access ID, that are described below. Both of the these access IDs are stored in an encrypted lockbox. The lockbox is associated with a particular host which prevents the lockbox from being copied from one host to another. Refer to the *EMC Solutions Enabler/Unisphere for VMAX version 8.0.1 Security Configuration Guide* for more information on the generating and storing access IDs.

Hardware-based access ID

By default, a host's access ID is derived from hardware characteristics of that host. On x86_64 (64-bit Intel/AMD) and IA 64 platforms, a network interface MAC address is used. On other platforms, different characteristics of the host, such as a processor identifier, are used.

Alternate access ID

An alternate access ID is generated at random or from a provided passphrase. This functionality is supported for all platforms but is strongly recommended on platforms where the access ID is derived from a network interface MAC address, such as x86_64 (64-bit), IA64, and BS2000 hardware platforms.

If the `SYMAPI_CLIENT_SIDE_ACCESS_ID` is enabled, then the `SYMAPI_ALTERNATE_ACCESS_ID` must be enabled. However, if the Alternate Access ID option is enabled on a host for the first time, along with the `SYMAPI_CLIENT_SIDE_ACCESS_ID` option and the client/server is set up, any command executed displays the following error: `Unable to obtain unique ID for host`. This is because the client is attempting to send the client access ID to the server, but there is no client access ID to send. Using the `symacl -unique -force` command in client/server mode to generate a client access ID is not permitted. To work around the issue, you must exit client/server mode on the client and execute the normal commands to generate a new access ID for the client. Again, this is only for a client that has never had the Alternate Access ID option enabled on it.

To forcibly generate a new alternate access ID, at the command line, using a supplied passphrase, enter:

```
symacl -unique -passphrase passphrase -force
```

Or from a file that contains the passphrase, enter:

```
symacl -unique -passphrase -file PathnameToFile -force
```

Note: In the client/server mode, the PathnameToFile is on the client host, although the alternate access ID is activated on the server.

Creating and modifying access control data

To create or change the Access Control information in the database, you need to enter the changes in a command file and commit the file to the access control database.

The command file format contains various command entries terminated with a semicolon (;). The command syntax is case insensitive, but any variable parameter entered must be case sensitive.

You can use the `symacl` command to perform the following operations:

- Create new access groups
- Add and remove access IDs to access groups
- Move an access ID from one group to another
- Create new access pools
- Add and remove devices to access pools
- Delete access pools and access groups
- Add ACEs to grant access
- Remove ACEs to deny access

Note: After Access Control changes are made, you must run a discover operation (`symcfg discover`).

Command file execution

To safely apply any of these Access Control actions (command file entries) to the access control database, perform the following progressive `symacl` operations on the command file:

1. Preview

   After you first create the command file, the *preview* operation verifies the syntax and accuracy of the contents of the entries in the command file.

2. Prepare

   The *prepare* operation (also before you commit) performs the preview checks, but also verifies the appropriateness of the requested access control modifications against the current state of the access control database in the array.

3. Commit

   The *commit* operation performs both the preview and prepare checks and then commits the contents of the command file to the Access Control database.

Note: It is not mandatory to execute a preview or prepare action prior to a commit. However, these actions can ensure that the commit action will not be rejected or can be used to debug the command file entries.

Minimum access control configuration

Typically, following the initial delivery and setup of an array at a site, EMC Customer Support needs to set some minimal access control parameters to enable you to begin administrative actions.

For a delivered setup, the array has its Access Control Database initialized with the creation of a Host Based Administrator and an access pin. This task is usually carried out by an EMC Engineer and is described in the following example:

Table 2 Initial access groups set

Access groups	ID names
AdminGrp	SunHost1 ACCPIN
UnknwGrp	unknown

The name `SunHost1` shown in the Table 2 (associated with `AdminGrp`) is the ID name that designates the machine from which you administer access controls. To perform prepare, commit, and release actions, an access ID (PIN) with the name `ACCPIN` was created in the delivered setup. This ID name is also used for the value setting of SYMCLI_ACCESS_PIN. For more information, see "Creating and managing access groups" on page 60.

Initial ACL setup

As shown in Table 3 for the delivered setup described in "Minimum access control configuration", access group AdminGrp has access to all devices (ALL_DEVS) and this group was granted both ADMIN and ALL permissions. Also in the initial setup, a group called UnknwGrp is established with permissions to all devices in the array. At the outset of system usage, this gives all unknown hosts BASE permissions to all devices (ALL_DEVS), until it becomes clear what restrictions should be established.

The initial `UnknwGrp` setup also grants ALL permissions to any devices not already assigned to an access pool (i.e.!INPOOLS).

Table 3 Initial delivered ACLs setup

Access groups	permissions (access types)	Access pools
AdminGrp	ADMIN ALL	ALL_DEVS
UnknwGrp	BASE	ALL_DEVS
UnknwGrp	ALL	!INPOOLS
EmcIntrnlGrp - used for internal array operations only	ALL	ALL_DEVS

Note: The access type ALL excludes ADMIN privileges.

For arrays running Enginuity 5773 through 5876, access control with Optimizer/FAST VP, requires a unique access control ID from EMC Customer Support that is internal to the array. This unique ID can be named EMC_SP and must be assigned to a group which is given Access Types BASE and OPTMZR to all devices (ALL_DEVS).

VLOGIX permission behavior

During initial setup of the system, the access group UnknwGrp (with Access Type ALL for NON-POOLED Devices) is present, as shown in Table 3 above. In this scenario, the VLOGIX privilege returns as true since you are granted access to all devices in the array. Initially, because no pools are present, the VLOGIX privilege is associated implicitly with all devices by this ACL.

Once you create an access pool and add a device to it, the VLOGIX privilege is no longer implicitly associated with all devices and, therefore, a check for the VLOGIX privilege would now fail. For the UnknwGrp to still have VLOGIX privilege, that privilege must be explicitly granted to the UnknwGrp and associated with ALL_DEVS.

Creating and managing access groups

Various sets of users tend to use the same applications that utilize common Solutions Enabler features from a given host. They typically require the same device resources and permissions of access to these shared devices. For this reason, hosts are registered in groups identified with a group name, which serves as a root for all ACEs in the group (see Figure 1 on page 56).

Access groups contain groups of access IDs and their ID names. Any ID and name must belong to just one group and are entered into the database together. For ease of management, it is highly recommended that you choose an access ID name that best associates with the particular host in use. For example, SunHost1 is more appropriate than a name such as JRSMITH. Once the group is created, the group name can be used to create access control entries (ACEs).

Setting up an access group

To set up an access control group, you need to create a group and then add the various access IDs of the hosts as members of that group. Initially, you can check to see if the host you are using has the administrative authority to make access control changes using the following command:

```
symacl list -v [-sid SymmID|ALL]
```

Create a group

For example, in the command file (`addnewgroups.cmd`), to create access groups named HR and Sales in array 12345, enter:

```
create accgroup HR;
create accgroup Sales;
```

You can now commit the file or wait until you have added access IDs to the group. To commit the file (addnewgroups.cmd) that executes the creation of the groups, enter:

```
symacl -sid 12345 commit -file addnewgroups.cmd
```

At this point, you may be prompted for your 4 - 12 character access PIN (if environment variable SYMCLI_ACCESS_PIN is not already set to this PIN value).

Add host access IDs to a group

For arrays running HYPERMAX OS 5977 or higher, the host access IdName is 1 - 31 characters. For arrays running Enginuity 5876 and lower, the user access IdName is 1 - 8 characters. In both cases the IdName is case sensitive, allows alphanumeric characters, plus underscore, and dash and no spaces. To add a host access ID to an access group, use the following command syntax in the command file:

```
add host accid Id name IdName to accgroup GroupName;
```

Note: The host access ID is encrypted; be sure to choose an *IdName* that you will remember.

For example, in the command file (`addnewgroups.cmd`), to add WinHost with an access ID `73900158-06174491-16225515` to the ProdB group, enter:

```
add host accid 73900158-06174491-16225515 name WinHost to accgroup
ProdB;
```

Also in this example, add UXNodeB with access ID 23300178-56078991-00665410 to the group GroupA, enter:

```
add host accid 23300178-56078991-00665410 name UXNodeB to accgroup
GroupA;
```

Note: To preserve access ID security for any host, host IDs are encrypted. Use `symacl -unique` to get this encrypted value.

Then you must commit the file (`addnewgroups.cmd`) to execute the addition of these IDs to their groups (and any create group entries if you have not already committed them):

```
symacl -sid 12345 commit -file addnewgroups.cmd
```

Note: After Access Control changes are made, you must run a discover operation (`symcfg discover`) to update your configuration.

User access IDs (PINs) for the AdminGrp

A user access ID is a PIN that allows a host to perform commit, prepare, or release operations to an array as the `AdminGrp`. When a host attempts a commit, prepare, or release operation as the `AdminGrp`, `symacl` will prompt for this PIN.

For arrays running HYPERMAX OS 5977 or higher, the user access IdName is 1 - 31 characters. For arrays running Enginuity 5876 and lower, the user access IdName is 1 - 8 characters. In both cases the IdName is case sensitive, allows alphanumeric characters, plus underscore, and dash and no spaces.

To add the user access ID for the `AdminGrp` access group, use the following command syntax in the command file:

```
add user accid Id name IdName to accgroup AdminGrp;
```

Note: The user access ID is encrypted; be sure to choose an *IdName* that you will remember.

For example, in the command file (`addnewgroups.cmd`), to add the administrative user access ID for `JOEPIN` as `1234PIN`, enter:

```
add user accid 1234PIN name JOEPIN to accgroup AdminGrp;
```

Note: User access IDs may only be set for the `AdminGrp` access group. Setting an ID for other groups will not return an error or prompt for a PIN in most scenarios.

If a user is prompted for a PIN, the user should enter the pin `1234PIN`. The `addnewgroups.cmd` file must be committed to execute the addition of these new IDs:

```
symacl -sid 12345 commit -file addnewgroups.cmd
```

Editing and managing an access group

Once access groups are established, you can remove access IDs or ACEs from a group, move IDs from one group to another, or delete an entire group.

Remove an access ID from a group

To remove an access ID from an access group, use the following syntax in the command file:

```
remove accid name IdName from accgroup GroupName;
```

For example, in the command file (`removeaces.cmd`), to remove user HRUser2 from the HR group, enter:

```
remove accid name HRUser2 from accgroup HR;
```

At some point, you must commit the file (`removeaces.cmd`) to execute the removal of this ACE from group HR.

Move an access ID to another group

To move an access ID from one access group to another, use the following command syntax in the command file:

```
move accid name IdName to accgroup GroupName;
```

For example, in the command file (`moveaces.cmd`), to move user HRUser1 to GroupA, enter:

```
move accid name HRUser2 to accgroup GroupA;
```

Then you must commit the file (`moveaces.cmd`) to execute the move of this ACE to GroupA.

Remove all ACEs from a group

To remove all ACEs from an access group, use the following syntax in the command file:

```
remove aces from accgroup GroupName;
```

For example, in the command file (`removeaces.cmd`), to remove all ACEs from the HR group, enter:

```
remove aces from accgroup HR;
```

At some point, you must commit the file (`removeaces.cmd`) to execute the removal of the ACEs from group HR.

Delete a group

To delete an access group from the database, use the following command syntax:

```
delete accgroup GroupName [remove_aces=true];
```

Note that when you need to delete a group, you either must have removed all ACEs from the group, or you can optionally remove all ACEs with the `[remove_aces=true]` option within this delete command. For example, in the command file (`deletegroup.cmd`), to delete group HR and any registered ACEs in that group, enter:

```
delete accgroup HR remove_aces=true;
```

Then commit the file (`deletegroup.cmd`) to execute the removal of access group HR.

Creating and managing limited access to access pools

Access pools are groups of devices controlled by access groups. When the various devices used by a host application must function as non-shareable resources, the target devices must be identified and assigned into an access pool for protection. Once an access pool is created, the pool can be a target to create access control entries (ACEs). More than one access group can access a pool with different permissions. For example, group AdminGrp might access PoolA with ALL permissions, while group HR could access the same PoolA with just BASE permissions.

Note: Once an access pool is created, any host in the access group UnknwGrp is denied access to the `symmask` and `symaccess` commands. In addition, when a host in UnknwGrp calls the `symmaskdb list database` or `symaccess list view -detail` commands, only its own devices will be returned in the list. To provide hosts in access group UnknwGrp with full access to the `symmask` commands, grant the access type VLOGIX (for ALL DEVS) to access group UnknwGrp. For more information on granting access types with full access to the `symmask` and `symaccess` commands, refer to "Grant permissions to an access group" on page 65.

Setting up access pools

You can create an access pool and add devices to a pool. Initially, you can check to see if the host you are using has the administrative authority to make access control changes using the following command syntax:

```
symacl list -v [-sid SymmID|ALL]
```

Create a pool

For example, in the command file (`addnewpool.cmd`), to create an access pool named PoolB in array 12345, enter:

```
create accpool PoolB;
```

You can now commit the file or wait until you have added devices to the pool. To now commit the file (`addnewpool.cmd`) to execute the creation of the pool, enter:

```
symacl -sid 12345 commit -file addnewpool.cmd
```

Add devices to a pool

To add the specific devices to a pool, use the following command syntax in the command file:

```
add dev StartDevName[:EndDevName] to accpool PoolName;
```

For example, in the command file (`addnewpool.cmd`), to assign device names 0A through 19 to PoolB, enter:

```
add dev 00A:019 to accpool PoolB;
```

Then you must commit the file (`addnewpool.cmd`) to the database:

```
symacl -sid 12345 commit -file addnewdevs.cmd
```

Editing and managing access pools

You can edit and manage the existing access pool information in the access control database. Devices can be removed from the access pool, and access pools can be deleted.

Note: After Access Control changes are made, you must run a discover operation (`symcfg discover`) to update your configuration.

Remove devices from a pool

To remove specific devices from a pool, use the following command syntax in the command file:

```
remove dev StartDevName[:EndDevName] from accpool PoolName;
```

For example, in the command file (`removedevs.cmd`), to assign device names 16 through 19 for removal in PoolB, enter:

```
remove dev 016:019 from accpool PoolB;
```

At some point, you must commit the file (`removedevs.cmd`) to execute the removal of devices 16 through 19 from PoolB.

Delete a access pool

You can delete an access pool and at the same time remove any associated access control entries (ACEs) with the specified pool. Note that all ACEs must be removed before you can delete a pool.

To delete an access pool and remove all ACEs, use the following command syntax in the command file:

```
delete accpool PoolName [remove_aces=true];
```

For example, in the command file (`deletepool.cmd`), to remove PoolB and any associated ACEs, enter:

```
delete accpool PoolB remove_aces=true;
```

At some point, you must commit the file (`deletepool.cmd`) to execute the removal of PoolB.

Creating and managing access control entries

Once you have created groups and access pools, you can now create access control entries (ACEs) that grant permissions to the groups and pools.

A group can have multiple permissions and pools, which requires an ACE for each.

Initially, you can check to see if the host you are using has the administrative authority to make access control changes using the following command syntax:

```
symacl list -v [-sid SymmID | ALL]
```

Grant permissions to an access group

Once you have created an access group, you need to determine and grant, to the access group, the permissions (*AccessType*) of access to a specified pool or to all the devices in the array (or all devices not in a pool) using the following command syntax:

```
grant access=AccessType to accgroup GroupName
   for accpool PoolName | ALL | NON-POOLED devs
```

The *AccessType* parameter specifies the permissions to the SYMCLI or SYMAPI features or functionality that is to be granted to selected devices.

Note: When issued against a VMAX 10K array with Enginuity 5875Q32011SR and higher, the `symacl` command can be used to grant SRDF permissions to a pool of devices (or all devices).

Available permissions

The possible permissions values are listed and described in Table 4.

Table 4 Permissions: AccessType (page 1 of 4)

Permissions (*AccessType*)	Description	SYMCLI commands affected
ADMIN [2]	Grants administrator privilege to grant/deny access control entries to hosts and users.	symacl ‹prepare, commit, release, list, show›
ADMINRD [2]	Grants read-only access to all access control information.	symacl ‹list, show›
ALL [2]	All possible access types granted except ADMIN and ADMINRD. Must be directed to ALL devices.	All

Table 4 Permissions: AccessType (page 2 of 4)

Permissions (*AccessType*)	Description	SYMCLI commands affected
BASE [4]	Allows the discovery of devices and to obtain states and statistics from the array (directors and devices).	Base component commands • symevent, symcfg, symaudit, symdg, symdev, symcg ‹list, show›, symdisk, symstat, symipsec (Policy priority list, Policy get, and IPSec statistics get operations) • symmigrate query and list • symlmf query, list, and show (for arrays running Enginuity 5875, 5876, or HYPERMAX OS 5977 with elicensing)
BASECTRL	Allows base control operations on devices and device groups.	symdg ‹controls›, symcg ‹controls›, symdev ‹controls›, symsg ‹controls›
BCV	Allows TimeFinder BCV and TF/Clone control and status operations.	symbcv, symmir ‹controls›, symclone ‹controls›
CACHCTRL	Allows Cache control operations concerning LRU partition management.	symqos ‹set LRU ›
CFGDEV	Allows powerful configuration control operations that manage various types of configuration changes on devices in the array.	symconfigure Allows the following types of operations: • Convert device configurations (BCV, SRDF, DRV) • Convert device configuration (changing mirroring) • Set device attributes • Set device emulation • Metadevice management

Table 4 Permissions: AccessType (page 3 of 4)

Permissions (*AccessType*)	Description	SYMCLI commands affected
CFGSYM [2]	Allows access to set array attributes, set port flags, and swap RA groups with `symconfigure` command. It also affects the `symmaskdb`, `symaccess`, and `symrdf` commands as specified in the next column. Must be directed to ALL devices.	symconfigure, symmaskdb, symrdf, symaccess Allows the following types of operations: • Spare management • SAVE device pool create/delete • SAVE device pool member management • Enable/disable SRDFA • Set matrix across the array • Add/remove dynamic SRDF groups • Change director supporting a dynamic SRDF group • Set the link limbo value for a dynamic SRDF group • Modify User Authorization settings via `symauth` • Converting thin device (adding SRDF) • Setting thin dev attributes • Bind/unbind thin device (not supported by symconfigure) • Configuring thin metadevices • Allocate/free thin device • Creating thin device pools • Adding devices to thin pools • Enabling/disabling devices in thin pools • Removing devices from thin pools • Deleting thin pools • symipsec policy add, replace/modify, and delete • Initialize the VCMDB • Convert the database type • Restore the database • Manage storage groups • Manage port groups • Manage initiator groups • Manage masking views
CHECKSUM	Allows array device Double Checksum operations.	symchksum ‹controls›
CREATEDV[2]	Allows the creation and deletion of array devices (part of `symconfigure`).	symconfigure • Create new devices • Configure disk space (create/map/mask/meta/attributes) • Delete devices • Create thin devices
DIRCTRL[2]	Allows you to take directors and their ports offline and online. Must be directed to ALL devices. Also allows you to set CHAP authentication.	symcfg ‹online/offline› • Online/offline RA directors • Online/offline front-end ports symconfigure • Set port flags and attributes symconnect, symaccess • CHAP authentication
ECC[1 - 2]	Allows the ECC Symmetrix agent to run on the requested host.	Not applicable

Table 4 Permissions: AccessType (page 4 of 4)

Permissions (*AccessType*)	Description	SYMCLI commands affected
OPTMZR	Allows user-configurable attributes that may affect the Optimizer and Symmetrix Migration behavior.	symoptmz ‹controls› symmigrate ‹controls› symfast ‹controls›
POWRPATH [1 - 2]	Access to PowerPath-directed devices in an SRDF consistency group. Must be directed to ALL devices.	Not applicable
QOS	Allows the execution of Quality of Service (QOS) performance control operations to manage copy priorities. Excludes LRU cache control functionality.	symqos ‹set pace›
RCOPY	Manage Open Replicator sessions.	symrcopy
RDF	Allows SRDF control and set operations.	symrdf ‹control›
RPA	For RecoverPoint splitter to operate correctly, you are required to grant BASE, BASECTRL, RPA, and CFGDEV access types.	Not applicable
SDDF	Allows the DeltaMark (Change Tracker) functionality that monitors track changes.	symchg
SDR	Allows mapping/unmapping of devices to directors/ports for the Symmetrix Disk Reallocation (SDR) feature, Device Masking, and Auto-provisioning Groups.	symconfigure • Manage CKD aliases • Map/unmap devices • Map/unmap thin device symmask, symmaskdb, symaccess • mapping and unmapping
SNAP	TF/Snapshot allows the creation and management of virtual copy sessions between a source device and multiple virtual (VDEV) target devices.	symsnap
VLOGIX [1 - 4]	Enables access to Device Masking devices and Auto-provisioning Groups.	symmask, symmaskdb, symaccess

1. See the appropriate product documentation for use of these access types.
2. These access types must be granted to either ALL devices or all NON-POOLED devices in an array.
3. This access type must be granted to ALL devices in an array.
4. BASE access allows the creation/modification of symaccess group types, which are not part of a view. VLOGIX access is required to create or manipulate items within a view

Before you can grant a group the permissions to any value-added, non-base feature, you must grant the group BASE permissions.

To grant BASE permissions to access group ProdB for all devices, use the following command in the command file (`grantrights.cmd`):

```
grant access=BASE to accgroup ProdB for ALL devs;
```

To grant BASE permissions to access group ProdB for poolPRODdevs, specify:

```
grant access=BASE to accgroup ProdB for accpool PRODdevs;
```

Note: When restricting a host to BASE access to a pool of devices, all devices mapped to the host will be visible along with any devices that are not mapped to the host, but are in the pool. To configure a host to see only devices that are in the pool, map only those devices to the host. In addition, remote (SRDF) arrays and their devices will not be discovered and the application registration database and the audit logs cannot be accessed since these may contain data relevant to other hosts.

To grant SRDF permissions to access group ProdB for pool PRODdevs, use the following command in the command file (`grantrights.cmd`):

```
grant access=RDF to accgroup ProdB for accpool PRODdevs;
```

Note: When issued against a VMAX 10K array with Enginuity 5875Q32011SR, the `symacl` command can be used to grant SRDF permissions to a pool of devices (or all devices).

To grant BCV and SDR permissions to access group HR for pool poolHR, use the following command in the command file:

```
grant access=BCV, SDR to accgroup HR for accpool poolHR;
```

Note that these command examples created three ACEs for BASE, BCV, and the SDR permissions, which now could be committed.

To make any nonregistered (*unknown*) host have BASE permissions to access group UnknwGrp for all devices in the array environment, use the following command in the command file:

```
grant access=BASE to accgroup UnknwGrp for ALL devs;
```

At some point, you must commit the file (`grantrights.cmd`) to execute the granting of permissions with these various command entries.

Remove permissions from an access group

To remove permissions for an access group, use the `remove access` command in the command file:

```
remove access=AccessType from accgroup GroupName
   for accpool PoolName | ALL|NON-POOLED devs;
```

At some point, you must commit the file (`grantrights.cmd`) to execute the change of permissions with these entries.

Obtaining access control information

You can check the current Access Control database to view the various groups, pools, and also the ACLs. Only if you have ADMIN or ADMINRD permissions, can you view all access objects in the access control database.

Using the list action when you do not have administrative permissions, you only see access objects that are associated with the access group to which your host belongs.

Listing access control information

You can list information about groups, pools, and ACLs with the following syntax:

```
symacl [-sid SymmID|ALL] [-h]
     list [-v]
     list [-accpool | -accgroup | -acl]
```

For example, the following command lists the access groups on array 0133:

```
symacl -sid 0133 list -accgroup
```

Showing access control information

To show detailed information about a specified group or pool, use the following syntax:

```
symacl [-sid SymmID | ALL]

        show accpool PoolName [-acl]

        show accgroup GroupName [-acl]
```

When using the show actions without administrative permissions, you only see access objects that are associated with the access group to which your host belongs.

For example, the following shows all the details about access group ProdB on array 0133:

```
symacl -sid 0133 show accgroup ProdB
```

Getting your host access ID

You can get the access ID for the host you are using. The following command will return a unique access ID in an encrypted format:

```
symacl -unique
```

The `-unique` option returns the access ID in the following segmented, 24-digit numeric form:

```
xxxxxxxx-yyyyyyyy-zzzzzzzz
```

For example:

```
12301558-94200021-00347892
```

Releasing pending locked sessions

During the processing of the access control command file, the `prepare` and `commit` actions are critical SYMCLI or SYMAPI operations that are considered access control *sessions*. In the event a host machine or application should abnormally fail and stop processing any prepare or commit access operation on its command file, you can abort this locked session from your current host or another host using the following command:

```
symacl release -sid SymmID
```

Note: If you are the security administrator and you intend to release a lock on the command file session, you must either set environment variable `SYMCLI_ACCESS_PIN` to your access ID or enter your PIN every time `symacl` prompts you.

Verifying a locked session

To verify an access controlled session is locked on any array, you can use:

```
symacl list -v
```

This command can tell you how long the session has been locked and who locked it.

Understanding access strategies

This section describes the access control strategies that can be applied in an access-controlled array environment. Several strategies can be considered for establishing or restricting access for a node or group of users or hosts to your environment. These strategies should be considered when you are setting up an access control environment for the first time.

Note: Remember, that after Access Control changes are made, you must run a discover operation (`symcfg discover`) to update your configuration.

Initially grant all permissions to everybody

The initial (delivered) strategy is to employ a default ID that controls all nodes not yet explicitly registered. This default ID can be used to grant a certain level or a minimal level of access for all unregistered nodes.

In Figure 2, when the array was initially delivered, a group named UnknwGrp was created for nonregistered hosts (with no ID).

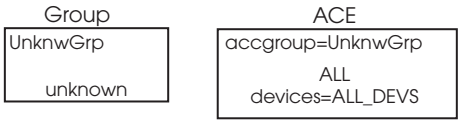


Figure 2 Example default configuration

A special *Default* access ID named `unknown` was added to the group granting all unknown hosts and users ALL permissions. Next, an ACE was created for group UnknwGrp granting them ALL permissions to all the array devices. In this scenario, all users and hosts can perform any of the SYMCLI command set operations.

Customizing, granting and denying default access

During Access Control implementation, the access IDs of all hosts that need to communicate to the array (for performing array operations using applications such as Solutions Enabler) are registered in the database and granted sufficient permissions to accomplish the functions they need to perform. Once the setup of Access Control is complete, one strategy could be to block or severely limit access for any other host not specifically defined. This is accomplished by adjusting the permissions of the default accgroup UnknwGrp.

The default access id UNKNOWN, affiliated with the accgroup UnknwGrp, is provided upon setup, to allow any host not specifically defined, to have a specific access. The accgroup UnknwGrp access rights can be tailored to provide certain hosts a specific access.

With special planning, the default access id UNKNOWN and the accgroup UnknwGrp can be removed which will prevent any access by a host, unless it is specifically defined with its own entries. This means that a host, unless specifically defined and associated with a group, will be denied any access.

For example, to remove the UnknwGrp, issue the following command from an admin host:

```
delete accgroup UnknwGrp remove_aces=true;
```

For example, if you need to revert back to the default settings concerning the UnknwGrp, you can issue the following commands from an admin host:

```
create accgroup UnknwGrp;
grant access=ALL to accgroup UnknwGrp for NON-POOLED devs;
grant access=BASE to accgroup UnknwGrp for ALL devs;
add default accid name UNKNOWN to accgroup UnknwGrp;
```

For example, to create the UnknwGrp with limited access, for example just BASE permission to all undefined hosts, use the following commands:

```
create accgroup UnknwGrp;
grant access=BASE to accgroup UnknwGrp for ALL devs;
add default accid name UNKNOWN to accgroup UnknwGrp;
```

Only one default accid is allowed. The one provided during setup is called UNKNOWN. Optionally, you can create a personalized default accid, by providing a name, and adding it to a Default type group using the following syntax:

```
add default accid name IdName to accgroup GroupName;
```

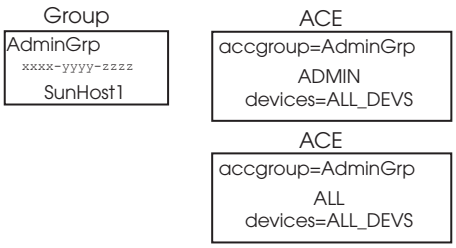
In the following example, after the default accid UNKNOWN is removed, a group named OtherGrp is created. A default ID of Others is assigned to the OtherGrp. The OtherGrp is then given permissions, which will be the limitations for all undefined hosts. These hosts are able to discover devices, obtain states and statistics from the array, with BASE permission granted to a default accid named OTHERS.

```
create accgroup OtherGrp;
grant access=BASE to accgroup OtherGrp for ALL devs;
add default accid name Others to accgroup OtherGrp;
```

Establishing an administrator

The delivered setup allows at least one host assigned with administrative (ADMIN) privileges to be designated the administrator of the information in the access control database.

In the following example, when the array was initially configured, an access group named AdminGrp was created. Then a UNIX workstation named SunHost1 with an encrypted access ID (of the form *xxxxxxxx-yyyyyyyy-zzzzzzzz*) was added to the AdminGrp group. (The access ID was obtained by running `symacl -unique`.)

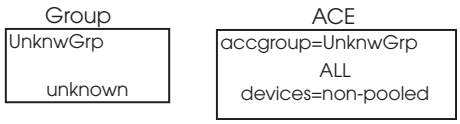


Next, two ACEs were created: one granting ADMIN permissions and one granting ALL permissions to group AdminGrp for all devices in the array.

Grant all permissions to the nonpooled devices

A second possible strategy once access controlled access pools have been established is to allow no restrictions on all the array devices that are not otherwise registered (*Nonpooled*) in any access pool.

In the following example, the ACE for the access group named UnknwGrp was modified to restrict access to only those devices not registered in an access control pool.



General absolute control strategy

A third strategy, which is more controlled, is to register access to only certain devices on an as-needed basis. In this scenario, the UnknwGrp group was removed. Therefore, a node must be known to the array, and only specific users/hosts in defined groups with limited or unlimited permissions have access to certain devices defined in their working pools.

Initial setup summary

Whatever strategy you apply during the initial building of the access control database, you need to survey all user needs, organize their limitations into groups, determine what devices to allocate to who, and what permissions are appropriate for each group. It is highly recommended to use the preview and prepare actions on your first major command file before you commit it; particularly if it is an extensive list. The preview and prepare will identify any coding errors or mistakes in the logic.

BCV setup strategy

To set up an access controlled environment for TimeFinder operations, you need to set up both the standard and BCV devices as follows:

1. Define your working access pool to contain both the standard and BCV SymDevNames.
2. For your group name, grant BASE permissions to access all devices.
3. For your group name, grant BCV permissions to the access pool holding the pairs.

TimeFinder/Snap setup strategy (Enginuity 5876 or lower)

To set up an access controlled environment for TimeFinder/Snap operations, you need to set up both the source and target devices and the SAVE devices as follows:

Source and target strategy:

1. Define your working access pool to contain both the source and target device's SymDevnames.
2. For your group name, grant BASE permissions to access all devices.
3. For your group name, grant SNAP permissions to the access pools holding the pairs.

SAVE device strategy:

1. Define your working access pool to contain the SAVE devices.
2. For your group name, grant BASE permissions to access the SAVE devices.
3. For your group name, grant CFGSYM permissions to create and delete the SAVE devices.
4. For your group name, grant SNAP permissions to manage the SAVE devices.

SRDF setup strategy

To set up an access controlled environment for SRDF operations, you need to set up both the local array and remote array. Because both arrays have their own access controlled database, you must:

1. With the ADMIN host, create an access control group for the local array. Then with an ADMIN local or remote host, define the same access control group name for the remote array.
2. With an ADMIN host, create an access pool defined with the R1 SymDevnames. Then with the ADMIN local or remote host, define an access pool with the R2 SymDevnames.
3. Grant BASE permissions for the group to access all devices on the R1 array. Then with an ADMIN host, grant BASE permissions for the group to access all devices on R2.
4. Grant SRDF permissions for the group to access the R1 access pool. Then with an ADMIN host, grant SRDF permissions for the group to access the R2 access pool.

Note: When issued against a VMAX 10K array with Enginuity 5875Q32011SR, the `symacl` command can be used to grant SRDF permissions to a pool of devices (or all devices).

RecoverPoint Splitter setup strategy (Enginuity 5876 or lower)

For the RecoverPoint splitter to operate correctly when array host-based access controls are enabled, you are required to grant BASE, BASECTRL, RPA, and CFGDEV access types.

For example, to grant BASE, BASECTRL, RPA, and CFGDEV permissions to access group ProdB for pool PRODdevs, specify:

```
grant access=BASE,BASECTRL,RPA,CFGDEV to accgroup ProdB for accpool
PRODdevs;
```

FAST/Optimizer setup strategy (Enginuity 5876 or lower)

To set up an access controlled environment for FAST operations, permission of OPTMZR must be granted to the host. Obtain the unique access control ID of the array from EMC Customer Support. This unique ID must then be added to access control and granted BASE and OPTMZR permissions.

Symmetrix Management Console/Unisphere setup strategy

Symmetrix Management Console (Enginuity 5876 or lower) and Unisphere software provide a GUI interface to query and manage an array. The GUI can be used to perform any operations to which the host has been granted rights.

Symaccess setup strategy

To enable symaccess control for a host, use the permission VLOGIX. This permission must be granted to ALL devices in the array. Note that the behavior of symaccess changes once access pools are created. Once access pools containing devices are created, VLOGIX type privileges are denied to groups unless the privilege is explicitly granted to any group needing access. Note that this includes the default UnknwGrp.

Backing up and restoring the access control database

It is a good practice to create a backup file of your existing access control database prior to making changes.

Create a backup file

You can create a backup of your current access control database using the following command syntax:

```
symacl -sid SymmID backup -file CommandFile
```

The backup operation saves the contents of the access control database in the file specified by the file option. The file must not previously exist. The backup file created is compatible for use with the `symacl` utility.

Note: The backup file contains encrypted versions of the unique IDs, therefore if you compare the values in the backup file to the original file used to create the database, they will be different.

Restore the backup file to the access control database

In the event a host machine or application makes undesirable changes to your access control configuration, you can utilize a backup file you have created to restore the previous configuration data to the access control database.

When you restore a backup using `symacl`, you use the `-restore` option as shown below:

```
symacl    -sid SymmID commit [-v|-noecho]
          -restore -file CommandFile
```

The restore operation replaces the contents of the access control database with the contents of the file specified by the file option. It eliminates whatever currently exists in the array configuration database and replaces it with the contents of the backup.

Note: The backup file contains encrypted versions of the unique IDs, therefore if you compare the values in the backup file to the original file used to create the database, they will be different.

CHAPTER 5
Grouping Devices

This chapter describes the benefits of grouping storage array devices, the types of groups, and how to create and modify them.

- Overview 78
- Device groups 78
- Storage groups 93
- Composite groups 124
- Using the GNS repository 138

Overview

In the Solutions Enabler environment several types of groups are commonly used monitor and control VMAX and DMX storage arrays. The type of group that you implement depends on your storage, host, and application environment. Grouping devices allows you to perform various functions including:

- Performing control operations on all devices in a group or on device pairs within the group that belong to one or more local arrays (**device groups**). Refer to "Device groups" on page 78 for information on creating and managing device groups.
- Managing groups of storage volumes that belong to a single array for mapping/masking, virtual LUN technology, FAST, and other base control operations (**storage groups**). Refer to "Storage groups" on page 93 for information on creating and managing storage groups.
- Managing groups of devices spread across multiple local arrays (**composite groups**). Refer to "Composite groups" on page 124 for more information on creating and managing composite groups.
- Ensuring remote data consistency (**SRDF consistency groups**). Refer to "SRDF consistency groups" on page 126 for more information on creating and managing consistency groups.

Device groups

A device group is a user-defined group comprised of devices that belong to a locally attached VMAX or DMX array. You can perform control operations on the group as a whole, or on the individual device pairs that comprise it. By default, a device cannot belong to more than one device group and all of the standard devices in a group must reside on the same array. However, if the options file parameter SYMAPI_ALLOW_DEV_IN_MULT_GRPS is enabled, you can add a device to multiple groups.

You can use device groups to identify and work with a subset of available devices; obtain configuration, status, and performance statistics on a collection of related devices; or issue control operations that apply to all devices in the specified device group.

A device group can belong to one or more composite groups. For more information, see "Composite groups" on page 124.

Group name services

In a default array environment, device group and composite group definitions are created through a locally-attached host. Upon creation, the group definition is stored in the host configuration database file. Therefore, only the host that created the group can see the group and control it. To perform control operations from another locally-attached host, the group definition must be manually copied to other hosts.

Optionally, you could enable the Group Name Services (GNS). GNS provides the optional ability to store device and composite group definitions in a shared repository located on each array, which then becomes automatically visible to all locally-attached hosts. This allows all GNS-enabled hosts to see the same group definitions across your environment, while sharing real-time updates to group definitions and configurations made by other hosts. For more information on GNS, refer to "Using the GNS repository" on page 138.

Names of device groups and devices

Device groups, as well as the devices in a device group, are assigned names that facilitate reference in a session. You assign a device group name at the time you create it. The name can have up to 31 characters and must be unique for a given configuration database.

When you add a device to a device group, it is given a logical name. This name allows you to refer to the device independently of its physical device name or array device name. The name can have up to 31 characters and must be unique within the device group. It is known only within the context of the device group to which the device belongs. You must use this logical name in any SYMCLI command with the argument `LdevName or ld`.

Device group types

You create a device group using one of the following types:

- REGULAR
- RDF1 (R1 and concurrent R11 devices)
- RDF2 (R2 and concurrent R22 devices)
- RDF21 (cascaded R21 devices)
- ANY (can contain a device mix of REGULAR, RDF1, RDF21 for an R1 site or RDF2, RDF21, RDF22 for an R2/R22 site)

Device lists

At the time of creation, a device group must be defined as type REGULAR, RDF1, RDF2, RDF21, or ANY, and may contain various device lists for standard, BCV, virtual (VDEV), and remote devices. A device is placed into a logical list when added to a device group. The following explains the device lists:

- **Standard device list** — You can group devices in a device group subject to the following restrictions:
  - You cannot mix SRDF and non-SRDF devices in the same device group *unless* you specify ANY for the type when creating a device group.
  - All SRDF devices in a given device group must belong to the same SRDF group or if concurrent SRDF, then two SRDF groups.
- **TimeFinder BCV device list** — You can associate Business Continuance Volume (BCV) regular devices and RDF1 BCV devices with a device group. You can perform BCV control operations on any BCV pair in the device group. Also, you can perform SRDF operations on just the associated RDF1 BCV devices. Refer to *EMC Solutions Enabler TimeFinder Family CLI Product Guide* for more information on adding BCV devices to a device group.
- **TF/Snap virtual device list** — A virtual device is a host-accessible device containing address pointers to the data stored on the source device or a pool of SAVE devices, which indicate where the copy session data is located in the physical storage. You can associate virtual devices (VDEV) paired with standard and BCV devices with any device group. In addition, TF/Snap control operations can be performed on any virtual device in a device group.

- **TF/Clone target list** — Target lists provide a mechanism for establishing source (SRC) and target (TGT) devices for TF/Clone operations. They can be created for both device groups and composite groups. A target list can contain various types of devices, including STDs, VDEVs, or BCV devices (based on a set of rules discussed in "Clone target restrictions" on page 81) and can use those devices as targets in clone operations. Remote target lists can also be created for remote operations.

  Note: VDEVs cannot be mixed with other types of devices in any target list. In other words, a target list cannot contain both VDEVs and STDs.

- **Gatekeeper device list** — You can *associate* one or more gatekeeper devices with a device group. SYMCLI uses the associated gatekeeper to issue requests to the array for control operations on the devices within the specified device group. You can *add* a standard device to a device group. However, the gatekeeper cannot be added to the device group, only associated with a device group. For more information about associating a gatekeeper device with a group, refer to the *EMC Solutions Enabler Installation Guide*.

  Note: A BCV device or a RDF2 device cannot be assigned as a gatekeeper, nor can a device that is a member of a device group be defined as a gatekeeper.

Creating a device group

A device group is created by first defining a named empty group of a specific type. Then, devices are added to the group. A newly-created device group is defined by the devices you add to it and each type of device list has its own set of restrictions.

Creating an empty device group

Device groups are created using the `symdg` command. When you create a device group, you assign it a name and choose it one of the group types explained in "Device group types" on page 79.

For example, to create a device group named `prod` whose members are operating as SRDF source devices, enter:

```
symdg -type RDF1 create prod
```

If a group type is not specified, the default group created is REGULAR.

Device restrictions

Some basic restrictions on the devices that you can add to a device group are based on the device type.

RDF1 and RDF2 type device restrictions

If you are adding an SRDF device to a device group, all devices in the device group:

- Must be SRDF devices.
- Must be either all source (RDF1 type) or all target (RDF2 type) devices.
- Must have the same SRDF group number.

- When there is a combination of R1 and R11 devices in a device group, the R1 and one of its R11 mirrors must have the same SRDF group number. This also applies for R2 and R22 devices.

RDF21 type device restrictions

The following device restrictions apply to all devices added to an RDF21 device group:

- All devices must be R21 STD devices. No mixture of STD device types is allowed.
- All R21 devices in a device group must maintain mirror consistency. All R1 mirrors must have the same SRDF group number and all R2 mirrors must have the same SRDF group number.
- Groups that have the same first SRDF group number, must have the same cascaded SRDF group number in a cascaded SRDF configuration.
- The existing rules for adding BCV, VDEV, and TGT devices to RDF21 groups still apply.

Virtual device restrictions

To add a device as a VDEV, the device must be defined as a virtual device. Virtual devices are VDEV, RVDEV and Hop-2 VDEV.

Clone target restrictions

STD, BCV and VDEV devices can be added to a target list. Below are the sets of device types allowed in a target list, although devices from only one set of devices is allowed in a given device group's target list at any given time:

- Non-SRDF STDs
- R1 STDs
- R2 STDs
- R1 + Non-SRDF Standards
- R2 + Non-SRDF Standards
- Non-SRDF BCVs.
- R1-BCVs
- R2-BCVs
- R1-BCVs + Non-SRDF BCVs
- R2-BCVs + Non-SRDF BCVs
- VDEVs (cannot mix VDEVs with other device types in the list)

By default, the logical device name (LdevName) for devices in a target list is TGT*xxx*. For devices in the remote target list, the default Ldevname is RTGT*xxx.*

ANY group type restrictions

The ANY group type allows a mix of non-SRDF and SRDF (R1, R11, R2, R22, and R21) devices in a single device group. It lifts the restrictions pertaining to the type of devices added to the device group but still follows the previous restrictions for SRDF devices. For example, all SRDF mirrors in a SRDF list must be of the same device type.

When there is a combination of R1 and R21 devices in a device group, the R1 mirror of the R21 devices must have the same SRDF group number. This also applies to R2 and R21 devices in a device group.

The following devices are allowed in a device group of ANY:

- All REGULAR devices
- All R1 and R11 devices
- All R2 and R22 devices
- All R21 devices
- A combination of REGULAR, R1, R11 and R21 devices
- A combination of REGULAR, R2, R22 and R21 devices

Note: A device group of any SRDF type can change its type because of an `symrdf` control operation. For example, an RDF1 composite group can change to an RDF2 when the device personalities are swapped. The `symrdf swap` operation does not change the type of an ANY device group.

SYMCLI provides various commands to add standard devices, virtual devices, or TF/Clone target devices to a device group. Once a device group is created, SYMCLI commands can be used to add a single device, all devices on a an array not yet assigned to a device group, or a list of devices from a file to that group. For more information on adding a list of devices from a file, refer to "Exporting and importing device lists" on page 86.

Adding devices to a device group

Use the `symdg` command to add a single device to a device group. Devices can be added by specifying either the physical device name (`add pd`) or the array device name (`add dev`). When adding virtual devices or target devices, specify the device name (*SymDevName*). Specify the interval and count options (`-i` and `-c`) to wait a predetermined time (interval) between attempts (count) to acquire an exclusive lock on the host database. The following is the syntax for adding a device:

```
symdg -g DgName [-offline] [-i Interval] [-c Count][-v]

        add pd PdevName [LdevName]

        add dev SymDevName [LdevName][-sid SymmID]
          [-rdf | -hop2 | -vdev | -tgt]
          [-rdfg GrpNum [-remote_rdfg RemoteGrpNum]
```

To add a single device using the physical device name `/dev/rhdisk32` to a device group named `prod`, enter:

```
symdg -g prod add pd /dev/rhdisk32
```

Optionally, you can assign a logical device name to the device. This logical device name can have up to 31 characters and must be unique within the device group. If a logical device name is not specified, one will be supplied automatically by SYMCLI.

To add device `00005` to a device group named `prod` and assign the logical device name `temp1`, enter:

```
symdg -g prod add dev 00005 temp1
```

Adding virtual devices

You can add a virtual device to the virtual device list (VDEV) of a device group or the remote virtual device list (RVDEV). When adding virtual devices from a local array, specify the `-vdev` option along with the device name. For example, to add virtual device 00005 to a device group named `prod1` and assign the logical device name `vdev1`, enter:

```
symdg -g prod1 add dev 00005 vdev1 -vdev
```

When adding virtual devices from a remote array (RVDEV), specify the `-vdev -rdf` option along with the device name. This targets the operation to the specified virtual device over SRDF links on the remote array. In the situation where you have concurrent SRDF, whereby there are two arrays on the remote side, you must specify the SRDF group number (`-rdfg GrpNum`) with `-rdf` option.

When adding virtual devices from an array two hops away, specify the `-vdev -hop2 -remote_rdfg RemoteGrpNum` option along with the device name. This targets the operation to the specified virtual device over SRDF links on the remote array two hops away.

Adding devices to the target list

You can add a device to the target device list (TGT) of a device group or the remote target device list (RTGT) for TF/Clone operations. You can add STD, SRDF, BCV and VDEV devices to the TGT, RTGT, and Hop2 TGT target lists.

When adding devices from a local array, specify the `-tgt` option along with the device name. For example, to add target device `00023` to a device group named `prod1` and assign the logical device name `tgt23`, enter:

```
symdg -g prod1 add dev 00023 tgt23 -tgt
```

When adding target devices from a remote array, specify the `-tgt -rdf` option along with the device name. This targets the operation to the specified target device over SRDF links on the remote array. In the situation where you have concurrent SRDF, whereby there are two arrays on the remote side, you must specify the SRDF group number (`-rdfg GrpNum`) with `-rdf` option.

When adding target devices from a array two hops away, specify the `-tgt -hop2 -remote_rdfg RemoteGrpNum` option along with the device name. This targets the operation to the specified target device list over SRDF links on the remote array two hops away.

For example, to add target device `00069` belonging to SRDF group `12` to a device group named `mywork` and assign the logical device name `tgt2`, enter:

```
symdg -g mywork add dev 00069 tgt2 -vdev -rdf -rdfg 12
```

For details on the types of devices that can be added to a device group's target list, refer to "Clone target restrictions" on page 81.

Logical device names for virtual devices

When a virtual device is added to a target list, its logical device name is assigned a VDEV default logical name according to these rules:

- VDEVs added to the target list are named TGTxxx.
- VDEVs added to the remote target list are named RTGTxxx.

- VDEVs added to the Hop-2 target list are named 2TGTxxx.

Adding ungrouped devices

Use `symdg` to add ungrouped devices to a device group. By default, this command adds all standard devices (or local virtual devices) to a device group. However, various optional criteria can be set to specify the types of devices to add to the specified device group.

Specify the interval and count options (`-i` and `-c`) to wait a predetermined time (interval) between attempts (count) to acquire an exclusive lock on the array host database on the local and/or remote arrays.

Use the following syntax to add ungrouped devices to a device group:

```
symdg -g DgName [-offline] [-i Interval] [-c Count][-v]
           [-sid SymmID]
           [-vdev | -tgt -rdf | -hop2
           [-rdfg GrpNum [-remote_rdfg RemoteGrpNum]]

        addall dev
          [-SA # | ALL] [-P #] [-N #]
          [-cap # [-captype mb | cyl]]
          [-sel_rdfg SelRdfgNum]
          [-devs SymDevStart:SymDevEnd | SymDevName
          ,SymDevStart:SymDevEnd | SymDevName . . .]
```

- All ungrouped devices from a specific ID (`-sid`).
- All virtual devices from a local or remote array (`-vdev -rdf -rdfg GrpNum`).
- All devices are to be added to the target list of the device group for TF/Clone operations on a remote array (`-tgt -rdf -rdfg GrpNum`).
- All virtual devices from a remote array two hops away (`-vdev -hop2 -remote_rdfg RemoteGrpNum`).
- All devices are to be added to the target list of the device group for TF/Clone operations on an array two hops away (`-tgt -rdf -hop2 -remote_rdfg RemoteGrpNum`).
- All devices visible to one or all front-end directors (`-SA <#|ALL>`).
- All devices visible to one or all front-end port number (`-P #`).
- The number of devices to add to the device group (`-N #)`.
- Any combination of device ranges and single devices, such as 225:22a,120,a5,7a0:7af (`-devs`), or that are of a specific capacity (`-CAP #`).
- Only SRDF devices belonging to that group number are added (`-sel_rdfg`).

Note: In the situation where you have concurrent SRDF, whereby there are two arrays on the remote side, you must specify the SRDF group number (`-rdfg GrpNum`) with `-rdf` option. When the `-hop2` option is specified, you must specify a `-remote_rdfg RemoteGrpNum`.

Logical names

The `addall` argument assigns the following logical names to the devices it adds: DEV001, DEV002,..., DEV*nnn*. You can use the `rename` argument to change these logical names after the devices have been added. Or, prior to calling this command, you can change the default logical device naming conventions using the SYMCLI_LDEV_NAMING environment variable. You can also set the SYMCLI_FULL_LDEVNAME environment variable to not truncate logical names too long to fit in the columns of the `symdg show` and `symcg show` output.

Restrictions

When using `addall`, you **cannot**:

- Mix devices from different arrays.
- Mix SRDF devices that have different SRDF group numbers.
- Add device(s) defined as a gatekeeper or BCVs.
- Add devices whose device type does not match the device group type.

Setting controls on Celerra devices

Use the following `-celerra` option to set the `rw_enble`, `write_disable`, `ready`, and `non_ready` controls on Celerra FBA devices in a device group:

```
symdg -g DgName [-noprompt] [-i Interval] [-c Count]
   [-bcv | -vdev | -tgt] [-star] [-celerra]
```

Listing devices in a device group

After adding devices to a device group, use the following syntax to list them:

```
symdg -g DgName list ld
```

Listing device groups

To list all device groups defined in the configuration database, enter:

symdg list

```
                          D E V I C E      G R O U P S
                Flags                          Number of
  Name    Type    VC    Symmetrix ID  Devs  GKs  BCVs  VDEVs  TGTs
  dgnocgs RDF1    YN    N/A              0    0     0      0     0
  dgincg  REGULAR YY    000194900341  2154    0   128      0     0

Legend:
  Flags:
       V(alid) DG     : Y = Valid, N = Invalid, - = N/A
       (In) C(g)      : Y = Contained by a CG, N = Not contained by a CG
```

This command returns a list of all device group names, group type, array ID, the number of standard, BCV, gatekeeper, virtual devices, and TF/Clone target devices (TGTs). It also tells you if the device group is valid and whether it is contained by a composite group.

Note: Device groups with names longer than 17 characters display with their first 17 characters followed by an asterisk (*). "Setting group names to display full names" on page 31 explains how to set the environmental variable to display longer group names.

Showing details about a device group

Use `symdg show` to display information about a specific device group. For example, the following output shows that device group `dgincg` is of type RDF1. This is only partial output for this command:

```
symdg show dgincg

Group Name:  dgincg

    Group Type                                  : ANY
    Device Group in GNS                         : No
    Valid                                       : Yes
    Symmetrix ID                                : 000194900341
    Group Creation Time                         : Tue Dec 1 8:6:9 2009
    Vendor ID                                   : EMC Corp
    Application ID                              : SYMCLI

    Number of STD Devices in Group              : 2154
    Number of Associated GK's                   :    0
    Number of Locally-associated BCV's          :  128
    Number of Locally-associated VDEV's         :    0
Number of Locally-associated TGT's           :    0
    Number of Remotely-associated VDEV's(STD RDF):    0
    Number of Remotely-associated BCV's (STD RDF):    0
    Number of Remotely-associated TGT's(TGT RDF) :    0
    Number of Remotely-associated BCV's (BCV RDF):    0
    Number of Remotely-assoc'd RBCV's (RBCV RDF) :    0
    Number of Remotely-assoc'd BCV's (Hop-2 BCV) :    0
    Number of Remotely-assoc'd VDEV's(Hop-2 VDEV):    0
    Number of Remotely-assoc'd TGT's (Hop-2 TGT) :    0
    Number of Composite Groups                  :    2
    Composite Group Names                       : cg110
                                                : cgregular
. . .
```

Exporting and importing device lists

You can save the list of devices from an existing group to a file on your host system, and this file can later be imported to create a device group. This can be useful if you delete the group and later wish to recreate it, or if you wish to import the group definition to another system.

Exporting a device list

You can remove all devices from a group but retain or export the list of devices in the group to a file on your host system using the following syntax:

```
symdg
          export <DgName> [-delete] [-file <FileName>]
             [[-rdf [-rdfg <GrpNum>]] | [-sid <SymmID>]]
             [-grpfile <GrpDbFileName>]

          exportall [-delete] [-file <FileName>]
            [[-rdf [-rdfg <GrpNum>]] | [-sid <SymmID>]]
            [-grpfile <GrpDbFileName>]
```

For example, to export the device group membership from group `prod2` to file `prod2list`, enter:

```
symdg export prod2 -f prod2list
```

To export an SRDF group, specifying the `-rdf` option uses the remote array ID and device names and changes the SRDF group type from R1 to R2 or vice versa. In addition, an SRDF group number can be specified as well (`-rdfg GrpNum`).

In addition, a text file that contains the details of all members of the existing device groups can be created using the `symdg exportall` operation. The device groups can later be recreated from this file using the `symdg importall` command.

Exporting and deleting a device group

The delete option allows you to export the device group membership to a file and delete the existing device group in the same operation. To reinstate the same group again, you can import this list (refer to "Importing device lists" on page 87) to the same or different device group.

Exporting device lists with -delete option

You can delete a group, but export the list of devices in the group to a file on your host system. For example, to export the device group membership to file `prod2list` from group `prod2` and then delete the group, enter:

```
symdg export prod2 -f prod2list -delete
```

For information on deleting a device group, refer to "Deleting a device group" on page 91.

Note: The `-rdf` option is not supported when exporting R21 device groups.

Importing device lists

You can add multiple devices to a new or existing device group by importing an existing file that contains a list of devices created by the export action.

For example, to create a device group named `prod2` from the file `prod2list`, enter:

```
symdg import prod2 -f prod2list
```

Typically, you import files that were previously exported (refer to "Exporting a device list" on page 86).

The `import` action creates the device group if the group name specified in the command does not already exist, or you can import to an existing group name that is partially populated. If you import to an existing group, the devices in the imported file are appended to the existing group membership.

In addition, you can recreate all device groups, using the `symdg importall` command, from data contained in a text file that was previously created using the `symdg exportall` command.

Note: For arrays running HYPERMAX OS 5977, Raid group members cannot be directly controlled, so exporting or importing details about specific Raid group members is not supported.

Renaming device groups

Device groups—and the logical device name of devices in a device group—can be renamed.

Device group

Use `symdg` to rename a device group. The new name can contain up to 31 characters and must be unique for the configuration database on which the device group is defined.

For example, to rename the device group `prod` to `prod_B`, enter:

```
symdg rename prod prod_b
```

If the device group is a member of one or more composite groups, specify the verbose option to view its associated composite groups, such as:

```
symdg rename prod prod_b -v

DG prod contained by CG cg1 was renamed
DG prod contained by CG cg14 was renamed
```

Devices

Use `symdg` to change the logical name of a device. The name can have up to 31 characters and must be unique within its device group.

For example, to rename the logical name of device `DEV003` to `TEMP3` in device group named `prod`, enter:

```
symdg -g prod rename ld DEV003 TEMP3
```

Moving devices

Use `symdg` to move one device, specifying the logical device name, from one existing device group to another existing device group. The source and destination device groups must be compatible types. Specify the interval and count options (`-i` and `-c`) to wait a predetermined time (interval) between attempts (count) to acquire an exclusive lock on the array host database and, for SRDF control operations, on the local and/or remote arrays. The following syntax is used to move an individual device:

```
symdg -g DgName [-h] [-offline] [-i Interval] [-c Count]
   move ld LdevName DestDgName [-force] [-rename]
```

For example, to move logical device `DEV003` from device group `prod` to device group `test`, enter:

```
symdg -g prod move ld DEV003 test
```

If there is a device within the destination device group with the same logical device name as device you wish to move into the destination device group, you must use the `-rename` option to avoid encountering an error. When this option is used, SYMCLI will rename the moved device to the next available logical device name as defined in the SYMCLI_LDEV_NAMING environment variable.

Moving all devices

When choosing to move all devices from one existing device group to another, you can move all devices, or use the additional options to focus the operation to devices that meet a specific criteria. By default, this operation only affects standard devices. Specify the

interval and count options (`-i` and `-c`) to wait a predetermined time (interval) between attempts (count) to acquire an exclusive lock on the array host database and, for SRDF control operations, on the local and/or remote arrays. However, the other option can be specified to only move specific device types as shown in the following syntax:

```
symdg -g DgName [-h] [-offline] [-i Interval] [-c Count][-v]

      moveall DestDgName [-force] [-rename]
         [-vdev | -tgt [-hop2] | -rvdev | -rtgt]
```

Other options include specifying devices mapped to a specific front-end (SCSI or Fibre) director number, (`-SA <#|ALL>`), devices mapped to a specific (SCSI or Fibre) director port number (`-P #`), devices of a specified capacity (`-CAP #`), total number of devices (`-N #`), or any combination of device ranges and single devices (`-devs`) to move. If `-devs` is not specified, all devices in the specified array are moved. The `-vdev` option adds only the virtual devices to the composite group. If the `-hop2` option is specified, it indicates that the specified device is two hops away. The `-tgt` option adds only the TGTs to the composite group. If the `-hop2` option is specified, it indicates that the specified device is two hops away.

The source and destination device groups must have compatible types.

For example, to move all virtual devices from device group `prod` to group `test`, enter:

```
symdg -g prod moveall -vdev test
```

Copying devices

Devices from an existing device group can be copied into another existing device group of compatible type. When the `copy` action is used, one standard device is copied from the specified source device group to the destination device group. The source and destination device groups must have compatible types. When the `copyall` action is used, copies of all standard devices are copied from the specified source device group to the destination device group. The source and destination device groups must have compatible types. When performing a `copyall` action, the types or number of devices that are included in the copy can be limited using the various filter options.

For a device to exist in multiple device groups (which a copy would do), requires the option file setting `SYMAPI_ALLOW_DEV_IN_MULT_GRPS = ENABLE`.

Use the following syntax to copy devices from one device group to another device group:

```
symdg -g DgName [-i Interval] [-c Count] [-v][-offline]

   copy ld LdevName DestDgName [-force] [-rename]

symdg -g DgName [-i Interval] [-c Count] [-v]
          [-offline] [-sid SymmID]
          [-SA # | ALL] [-P #] [-N #]
          [-cap # [-captype mb | cyl]]
          [-sel_rdfg SelRdfgNum]
          [-devs SymDevStart:SymDevEnd | SymDevName
               [,SymDevStart:SymDevEnd | SymDevName...]]
          [-R1 | -R2 | -R21 | -noRDF]

   copyall DestDgName [-force] [-rename]
      [-vdev | -tgt [-hop2] | -rvdev | -rtgt]
```

Copying devices from a device group to a storage group

Devices from an existing device group can be copied (or added) to a storage group. The copied devices remain in the device group (*DgName*) and are added to the destination storage group (*SgName*).

Use the following syntax to add devices from an existing device group to a storage group:

```
symdg dg2sg DgName SgName [-bcv|-vdev|-tgt]
```

If the storage group does not exist, it will be created. If optional device types are not specified, only standard devices will be added.

To add devices from a device group named `prod to a storage group named prod_2`, enter:

```
dg2sg prod prod_2
```

All standard devices in device group `prod` were added to storage `group prod_2`.

Removing devices

Use `symdg` to remove a device from a device group. Devices can be removed individually using the `remove` argument, or all devices can be removed from a device group using `rmall`. By default, the `remove` and `rmall` arguments will affect standard devices only. However, the `-vdev` option can be specified to remove just the virtual devices.

For example, to remove logical device `DEV003` from a device group named `prod`, enter:

```
symdg -g prod -force remove ld DEV003
```

In this example, the `-force` option removes the device regardless of its BCV state.

Note: If you remove the only member of a device group, the device group is not automatically deleted. You must execute `symdg delete` to explicitly delete the device group, as described in the "Deleting a device group" on page 91.

Removing all devices

When choosing to remove all devices from an existing device group, you can use the additional options shown in the following syntax:

```
symdg -g DgName [-i Interval] [-c Count] [-v]
          [-offline] [-sid SymmID]
          [-SA # | ALL] [-P #] [-N #]
          [-cap # [-captype mb | cyl]]
          [-sel_rdfg SelRdfgNum]
          [-devs SymDevStart:SymDevEnd | SymDevName
               [,SymDevStart:SymDevEnd | SymDevName...]]
          [-R1 | -R2 | -R21 | -noRDF]

      rmall [-force]
          [-vdev | -tgt -rdf [-rdfg GrpNum] | -hop2]
```

- Options include specifying devices mapped to a specific front-end (SCSI or Fibre) director number, (`-SA <#|ALL>`), devices mapped to a specific (SCSI or Fibre) director port number (`-P #`), devices of a specified capacity (`-CAP #`), or any combination of device ranges and single devices (`-devs`) to remove, or local and remote virtual and target devices (`[-vdev | -tgt -rdf [-rdfg GrpNum] -hop2]`).

Deleting a device group

Use the following syntax to delete a device group:

```
symdg delete DgName [-force]
```

For example, to delete device group named `prod`, enter:

```
symdg delete prod
```

If the device group is a member of one or more composite groups, specify the verbose option to view its associated composite groups, such as:

```
symdg delete prod -force -v
DG prod was removed from CG cg1
DG prod was removed from CG cg14
```

Note: Deleting populated device groups or a device group of a composite group requires the use of the `-force` option.

Performing operations on device group devices

Use the command syntax shown below to perform the following control operations on devices within a device group:

- Change the state (ready, not ready) of a device
- Set or reset the hold bit on a device
- Set or unset persistence on a device
- Sets or unsets Geometry Compatible Mode (GCM)

The following functionality is only supported on arrays running Enginuity 5876 or lower:

- Compress or uncompress a thin device
- Bind, unbind, rebind, start or stop allocations, free, and reclaim thin devices

Note: For Enginuity versions 5876 or lower, free and reclaim actions are blocked for any device which is the source or target of a Clone, Emulation, or Snap session.

- Set the Optimized Read Miss mode (ORM)

The command syntax is as follows:

```
symdg -dg DgName [-noprompt] [-i Interval] [-c Count]
          [-bcv [-hop2] | -rbcv | -brbcv |-rrbcv |
           -vdev [-hop2] | -rvdev | -tgt [-hop2] |
           -rtgt] [-rp] [-star] [-celerra]

        not_ready [LdevName [LdevName ...]]

        ready [LdevName [LdevName ...]]

        hold [LdevName [LdevName ...]]

        unhold [LdevName [LdevName ...]]

        compress [LdevName [LdevName ...]] [-stop]
```

```
uncompress [LdevName [LdevName ...]] [-stop]

bind [LdevName [LdevName...]] -pool <PoolName>

rebind [LdevName [LdevName...]] -pool <PoolName>

unbind [LdevName [LdevName...]]

allocate [LdevName [LdevName...]] [-persistent | -stop]

free [LdevName [LdevName...]] [-stop]

reclaim [LdevName [LdevName...]] [-persistent | -stop]

set -persistent [LdevName [LdevName ...]]

unset -persistent [LdevName [LdevName ...]]

set -orm [on | off | system] [LdevName [LdevName...]]

set -gcm [<LdevName> [<LdevName> ...]] [-symforce]

unset -gcm [<LdevName> [<LdevName> ...]] [-symforce]
```

Note: "Showing details about thin devices" on page 465 provides information on compressing and uncompressing thin devices.

The operations can be directed at all devices in a device group. By default the actions only apply to the standard devices in the group. If a control operation needs to be performed on devices in a specific list, the appropriate qualifier needs to be used (such as, `-bcv`, `-rbcv`, etc.).

For more information on actions for thin devices (bind, unbind, rebind, allocate, etc.) refer to "Performing actions on thin devices (Enginuity 5773 through 5876)" on page 284.

For arrays running HYPERMAX OS 5977, you can use the `set -host_id` option to modify the host ID of devices in a device group. Arrays are assigned a new device host ID format, which is a Federated ID that is globally unique and transportable between storage systems. When creating device groups, you can choose a host ID type that is either the new Federated ID format (`native`) or an ID compatible with the previous ID format (`compatible`) that is a non-portable ID value only unique within the array. The following restrictions apply:

- When `compatible` is specified, if the device number of any of the devices in the storage group is greater than 0xFFFF, the operation will fail.
- The operation will fail if any of the devices are online.
- The operation will fail if any of the devices are visible to a host through either mapping or masking.
- The operation will fail if `native` is specified and the emulation type of the device is AS/400_D910_099 on an array running HYPERMAX OS 5977.

For arrays running Enginuity 5876Q22013SR, the `symdg` command allows you to set the Optimized Read Miss mode to Off or System Managed for devices in a device group. By default, the action is only applied to standard devices in the group. If the action needs to

be performed on devices in a specific list, the appropriate qualifier needs to be used (such as, `-bcv`, `-rbcv`, etc.). "Optimized Read Miss mode (Enginuity 5876 only)" on page 439 provides additional information on this feature.

Storage groups

Storage groups are a collection of devices stored on the array that are used by an application, a server, or a collection of servers. Storage groups are used to present storage to hosts in masking/mapping, Virtual LUN Technology, FAST, and various base operations. The SYMCLI `symsg` command allows you to create and manage these storage groups.

Note: Chapter 6, "Fully Automated Storage Tiering," describes how storage groups are used to perform FAST-related operations.

Restrictions

The following general restrictions apply to all storage groups containing only devices. This applies to storage groups containing child storage groups only, not cascaded storage groups containing both parent storage groups with child storage groups:

- GNS does not support storage groups. Storage groups are saved in a special area on the array.
- A single array can contain a maximum of 8192 storage groups.
- Each storage group can contain a maximum of 4096 devices.
- Storage groups are supported only on VMAX arrays running Enginuity version 5874 to 5876 or HYPERMAX OS 5977.
- Storage group names can be up to 64 characters in length. Names must begin with an alphanumeric character and may contain embedded hyphens and underscore characters. Names are not case sensitive. Therefore, two storage groups named test and Test are not allowed.
- For VMAX arrays running Enginuity 5874 to 5876, a device cannot be added to a storage group associated with a FAST policy if the device already exists in another storage group that is associated with a FAST policy.
- Diskless devices are not permitted in storage groups.
- Logical device names are not supported by storage groups.

Additional usage restrictions for storage groups are described in the following sections:

- "Restrictions on parent and child storage groups" on page 102
- "Restrictions for storage groups with defined host I/O limits" on page 105

Working with storage groups

This section explains how to use SYMCLI commands to create a storage group and how to add and remove devices from a group.

Creating storage groups

The `symsg create` command allows you to create an empty storage group by defining a named group on the array. You can then add devices to the group. You can optionally set the front-end bandwidth, IOPs range and Host IO Limit dynamic distribution setting for the storage group. For HYPERMAX OS 5977 you can also set the Service Level Objective, Storage Resource Pool, and workload and RDF coordination settings.

Use the following syntax:

```
symsg -sid <SymmID> [-i <Interval>] [-c <Count>] [-v]

        create <SgName>
          [-bw_max <MBperSec>]
          [-iops_max <IOperSec>]
          [-dynamic <NEVER | ALWAYS | ONFAILURE>]
          [-slo <SLO Name> [-wl <Workload Name>]]
          [-srp <SRP Name>]
```

Where:

`-bw_max` — Limits the front-end bandwidth of the devices in the storage group. The valid range for bandwidth is from 1 MB/Sec to 100,000 MBs/sec.

`-iops_max` — Limits the I/Os per second of the devices over a set of director ports. The valid range for IOPs is from 100 IOs/sec to 100,000 IOs/sec but must be specified in units of 100 IO/Sec.

`-dynamic` — Specifies the Host IO Limit dynamic distribution setting for the storage group as follows:

- `NEVER`: The Host IO Limits for a storage group are never dynamically redistributed (static).
- `ALWAYS`: The Host IO Limits for the storage group are always dynamically redistributed.
- `ONFAILURE`: The Host IO Limits for the storage group are dynamically redistributed only upon failure of a front-end port.

Note: "Setting Host I/O Limits for storage groups" on page 104 describes how to use the `bw_max`, `iops_max`, and `dynamic` options to set Host I/O limits.

`-slo` — (HYPERMAX OS 5977 or higher) Specifies a Service Level Objective on a storage group. You can optionally set a workload when setting an Service Level Objective by supplying the `-wl` option.

`-srp` — (HYPERMAX OS 5977 or higher) Specifies a Storage Resource Pool on a storage group.

Note that the Service Level Objective and Storage Resource Pool name parameters function as follows:

- If you do not specify either *SLO Name* or *SRP Name*, then the storage group will be created with no Service Level Objective and no Storage Resource Pool and will *not* be FAST-managed.
- If you specify both *SLO Name* and *SRP Name*, the specified Service Level Objective and Storage Resource Pool will be used and the storage group will become FAST-managed.
- If you specify only *SLO Name*, the specified Service Level Objective and a default Storage Resource Pool for the emulation type of the devices in the storage group will be used and the storage group will become FAST-managed.
- If you specify only *SRP Name*, an Optimized Service Level Objective will be used with the storage group and the storage group will become FAST-managed.

Restrictions

- Storage Admin. permission is required.
- The command fails if the specified Service Level Objective, workload, or Storage Resource Pool does not exist.
- The command will fail if the specified Service Level Objective cannot be supported based on the Storage Resource Pool that is being used by the storage group.
- When setting a workload on an group, the command fails if a Service Level Objective is not set.

Creating new devices and adding them to storage groups

Use the `symconfigure create dev` command to create and add a device to an existing storage group during device creation.

Note: (HYPERMAX OS 5977 only) If a storage group is not specified or if the specified storage group is not FAST-managed, the device is created using the default Storage Resource Pool for the device's emulation type and an Optimized Service Level Objective.

Use the following syntax to add a device to a storage group:

```
create dev count=<n>,
     size = <n> [MB | GB | CYL],
     emulation=<EmulationType>,
     config=<DevConfig>
     [, remote_config=<DevConfig>,
      . . .
     [, sg=<SgName> [,remote_sg=<SgName>]],
```

For example, to create a device and add it to a storage group named “SG_Finance”, enter:

```
symconfigure -sid 230 commit -cmd "create dev count = 2, size = 1000 cyl,
            emulation = fba, config = TDEV, SG = SG_Finance;"

Execute a symconfigure operation for symmetrix '000197100230' (y/[n]) ? y

A Configuration Change operation is in progress. Please wait...

    Establishing a configuration change session...............Established.
    Processing symmetrix 000197100230
    {
          create dev count = 2, size = 1000 cyl, emulation = fba,
          config = TDEV, SG = SG_Finance;
    }

    Performing Access checks..................................Allowed.
    Terminating the configuration change session..............Done.

The configuration change session has successfully completed.
```

Restrictions

- Storage Admin. permission is required.
- The command fails if the specified storage group does not exist.
- The command fails if attempting to add an FBA device to a FAST-managed storage group that contains CKD devices or to add a CKD device to a group that contains FBA devices. This includes previous device create operations that add devices to the same storage group during one configuration change session.
- The command fails if attempting to add devices to a storage group that contains encapsulated devices. This includes previous device create operations that add devices to the same storage group during one configuration change session.

Adding existing devices to storage groups

Use the `symsg add dev` command to add devices to a storage group.

To add devices, use the following form:

```
symsg -sg SgName -sid SymmID [-i Interval] [-c Count]
   add dev SymDevName
```

To add multiple devices, or devices in a range or a file, use the following form:

```
symsg -sg SgName -sid SymmID [-i Interval]
          [-c Count] [-v]
          [-SA <# | ALL>] [-P #] [-N #]
          [-cap # [-captype <mb> | <cyl>]]
          [-devs <SymDevStart:SymDevEnd | SymDevName
            [,<SymDevStart:SymDevEnd | SymDevName>...]> |
           -file DeviceFileName [-tgt]>]
addall [pd | devs]
```

- To add a single device to a storage group named `prod` on array 123, enter:

  `symsg -sid 123 -sg prod add dev 30`

  Storage device 30 is added to storage group `prod`.

- To add all devices that are primarily visible from the host (mapped) to a storage group named `prod` on array 123, enter:

  `symsg -sid 123 -sg prod addall pd`

  All storage devices with physical (host) device names are added to storage group `prod`.

- To add a range of physical devices to a storage group named `prod` on array 123, enter:

  `symsg -sid 123 -sg prod addall pd -devs 30:3F`

  All storage devices within range 30:3F are added to storage group `prod`.

- Although the `-devs` option implies you can enter a single range or a list of comma delimited devices, the option supports a free-form combination of entries. For example:

  `symsg -sid 207 -sg sg1 -devs  64:105,22a,505,600:605,0700`

- To add all devices listed in a text file named storgrp_a.txt to a storage group named `prod` on array 123, enter:

  `symsg -sid 123 -file storgrp_a.txt -sg prod addall`

  All storage devices within text file `storgrp.txt` are added to storage group `prod`. Device text files can be in either a one or two column format, specifying source and target devices. If the `-tgt` option is specified, only devices from the second column are added to the storage group.

Note: Any attempt made to add a device belonging to a storage group that is part of a masking view to a second storage group fails.

Restrictions

- Storage Admin. permission is required.
- If the storage group is FAST-managed, the command fails if the device is already in another storage group that is FAST-managed.
- If the storage group is FAST-managed, the command fails if adding encapsulated devices.

Adding child storage groups to existing storage groups

Use the `symsg add sg` command to add a child storage group to a storage group.

To add devices or storage groups, use the following form:

```
symsg -sg SgName -sid SymmID [-i Interval] [-c Count]
   add sg SgName [,SgName2, SgName3, ..., SgNameN]
```

Restrictions

- Storage Admin. permission is required.
- The command fails when adding a child storage group to a FAST-managed storage group.

Changing properties of existing storage groups

Use the `symsg set` command to modify the Service Level Objective, Storage Resource Pool, workload and RDF coordination settings for an existing storage group.

Use the following syntax:

```
symsg -sg <SgName> -sid <SymmID> [-i <Interval>]
          [-c <Count>]

        set <[-bw_max <MBperSec> | NOLIMIT ]
             [-iops_max <IOperSec> | NOLIMIT ]
             [-dynamic <NEVER | ALWAYS | ONFAILURE>]>
             [-slo <SLO Name> [-wl <Workload Name>] |-noslo]
             [-srp <SRP Name> | -nosrp]
```

Where:

`-bw_max` — Limits the front-end bandwidth of the devices in the storage group. The valid range for bandwidth is from 1 MB/Sec to 100,000 MBs/sec.

`-iops_max` — Limits the I/Os per second of the devices over a set of director ports. The valid range for IOPs is from 100 IOs/sec to 100,000 IOs/sec but must be specified in units of 100 IO/Sec.

`-dynamic` — Specifies the Host IO Limit dynamic distribution setting for the storage group as follows:

- `NEVER`: The Host IO Limits for a storage group are never dynamically redistributed (static).
- `ALWAYS`: The Host IO Limits for the storage group are always dynamically redistributed.
- `ONFAILURE`: The Host IO Limits for the storage group are dynamically redistributed only upon failure of a front-end port.

Note: "Setting Host I/O Limits for storage groups" on page 104 describes how to use the `bw_max`, `iops_max`, and `dynamic` options to set Host I/O limits.

`-slo` — Modifies the Service Level Objective on a storage group. You can optionally set a workload when setting an Service Level Objective by supplying the `-wl` option and the name of the workload. If you do not specify a workload, then the workload `<none>` is assigned.

`-noslo` — Specifies a Service Level Objective to be removed from a storage group.

`-srp` — Specifies a Storage Resource Pool on a storage group.

`-nosrp` — Specifies a Storage Resource Pool to be removed from a storage group.

The Service Level Objective and Storage Resource Pool name parameters are optional and function as follows:

To set a user-supplied Service Level Objective on a storage group

Specify the `-slo` option and the desired SLO name. You can optionally set a workload when setting a Service Level Objective by supplying the `-wl` option with the desired workload name. If no workload is specified, then the workload `<none>` is assigned. If the storage group does not have a Storage Resource Pool set, the system default Storage Resource Pool for the emulation type of the devices in the storage group (FBA or CKD) will be used and the group will become FAST-managed.

Note: If your VMAX environment uses mixed FBA and CKD emulation types, then both default Storage Resource Pools will exist: one for FBA and one for CKD. If you add an FBA device to the storage group, then it will be added to the Storage Resource Pool for FBA. If you attempt to add CKD devices to the storage group with FBA devices, you will be blocked from doing so.

To remove a Service Level Objective from a storage group

Specify the `-noslo` option. This also removes any workload that was assigned for the Service Level Objective. If the storage group has a Storage Resource Pool, then the storage group will be assigned an Optimized Service Level Objective. Otherwise, there will be no Service Level Objective or Storage Resource Pool for the storage group and the group will no longer be FAST-managed.

To set a user-supplied Storage Resource Pool on a storage group

Specify the `-srp` option with the desired Storage Resource Pool name. If the storage group does not have a Service Level Objective set, then an Optimized Service Level Objective will be used with the storage group and the group will become FAST managed.

To remove a Storage Resource Pool from a storage group

Specify the `-nosrp` option. If the storage group has a Service Level Objective set, then the system default Storage Resource Pool for the emulation type of the devices in the storage group will be used. Otherwise, there will be no Storage Resource Pool or Service Level Objective for the storage group and the group will no longer be FAST-managed.

To remove both the Service Level Objective and the Storage Resource Pool from a storage group

Specify both the `-noslo` and the `-nosrp` options. The storage group will no longer be FAST-managed.

To change the workload on an storage group without changing its assigned Service Level Objective

Specify both the `-slo` option with the current Service Level Objective name and the `-wl` option with the new workload name.

To remove the workload on storage group while retaining the current Service Level Objective

Specify the `-slo` option with current Service Level Objective name and omit the `-wl` option. This causes the workload `<none>` to be assigned. The storage group will remain FAST-managed.

Restrictions

- Storage Admin permission is required.
- The command fails if the user-supplied storage group, Service Level Objective, workload, or Storage Resource Pool does not exist.
- The command fails if the user-supplied Service Level Objective cannot be supported based on the Storage Resource Pool that is being used by the storage group.
- The command fails if setting a Service Level Objective or a Storage Resource Pool to a parent storage group.
- When setting a Service Level Objective or a Storage Resource Pool, the command fails if any device in the storage group is already in another storage group which is FAST-managed.
- When setting a Service Level Objective or a Storage Resource Pool, the command fails if the storage group contains encapsulated devices.
- When setting a workload on a storage group, the command fails if the storage group does not have a Service Level Objective or a Service Level Objective is not being set. It will also fail if a Service Level Objective set on the storage group is being removed.

Converting between standalone storage groups and cascaded groups

Starting with Solutions Enabler V8.0.1, two CLI commands allow you to convert both ways between standalone storage groups and cascaded groups.

Converting a standalone storage group to a cascaded group

The `symsg convert -cascaded` CLI command allows the non-disruptive conversion between a standalone storage group and a cascaded storage group consisting of a parent storage group and a single child storage group. If the standalone storage group has a Host IO Limit, then you must specify if after the conversion the limit will be set on the parent or the child storage group. The standalone storage group has Host I/O Limits set if a limit to either `bw_max` or `iops_max` was configured on the group.

Use the following syntax to convert a storage group to a cascaded group:

```
symsg -sid <SymmID> [-i <Interval>] [-c <Count>] [-v]

   convert -cascaded <SgName> <ChildSgName> [-host_IO <on_parent | on_child>]
```

Restrictions:

- The following security privileges are required to execute this command:
  - Required Access type: BASE (if the storage group is not in a masking view)
  - Required Authorization Rights: VLOGIX (if the storage group is in a masking view)
- The command will fail if the user-supplied storage group does not exist.
- The command will fail if the supplied child storage group already exists.
- The command will fail if the supplied storage group is not standalone (that is, it is either a parent or child in an existing cascaded storage group configuration).
- The command will fail if the supplied storage group has a Host IO limit defined and the host_IO option was not given.

Converting a cascaded group to a standalone group

The `symsg convert -standalone` CLI command allows the non-disruptive conversion between a cascaded storage group consisting of a parent storage group and a single child storage group and a standalone storage group. If either the parent or the child storage group has a Host IO limit defined, it will be set on the standalone storage group. But if both parent and child storage groups have a Host IO limit, you must supply the host_IO option. A storage group has Host I/O Limits set if a limit to either `bw_max` or `iops_max` was configured on the group.

Use the following syntax to convert a cascaded group to a standalone storage group:

```
symsg -sid <SymmID> [-i <Interval>] [-c <Count>] [-v]

   convert -standalone <SgName> [-host_IO <keep_parent | keep_child>]
```

Restrictions:

- The following security privileges are required to execute this command:
  - Required Access type: BASE (if the SG is not in a Masking View)
  - Required Authorization Rights: VLOGIX (if the SG is in a masking View)
- The command will fail if the user-supplied storage group does not exist.
- The command will fail if the supplied storage group is either standalone or a child storage group is in a cascaded storage group configuration.
- The command will fail if the supplied storage group is a parent storage group that contains more than one child storage group.
- The command will fail if both the supplied storage group and its child storage group have a Host I/O Limit defined and the host_IO option was not given.

Adding cascaded storage groups (Enginuity 5876 and HYPERMAX OS 5977 only)

Solutions Enabler with Enginuity 5876 or HYPERMAX OS 5977 provides the capability for storage groups to contain other storage groups (cascaded storage groups). This cascading of storage groups allows for individual FAST policies for the storage groups containing devices and a masking view for the storage group containing other storage groups. The storage group, containing only devices, that is contained within a parent storage group is referred to as the child storage group.

You can add child storage groups individually to a parent storage group using the `symsg add sg` command.

Use the following syntax for adding child storage groups to a parent storage group:

```
symsg -sg SgName -sid SymmID [-i Interval]
     [-c Count] [-v] [-celerra] [-rp] [-ckd]

   add sg SgName1[,SgName2,SgName3,…,SgNamen]
```

Note: With Enginuity 5876Q22013SR or HYPERMAX OS 5977, the `symsg add` command allows a storage group with Host I/O limits set to be added to a parent storage group that is in a provisioning view using a port group that has FCoE ports.

Restrictions

The following additional restrictions apply to `symsg add sg` operations:

- You cannot add a device to a storage group associated with a FAST policy, if the device already exists in another storage group that is also associated with a FAST policy.
- Any attempt made to add a storage group that is part of a masking view to a second storage group fails.
- If the Host I/O limits (either `-bw_max` or `-iops_max`) on the child storage group that is being added is greater than what is configured on the parent storage group, then the operation will fail.

Restrictions on parent and child storage groups

The following restrictions apply to cascaded storage groups. This applies to storage groups containing other storage groups (a parent storage group and children storage groups):

- Support for cascaded storage group is only provided on arrays running Enginuity 5876Q42012SR or HYPERMAX OS 5977.
- Only a single level of cascading is permitted. A parent storage group may not be a child of another storage group.
- Storage groups can only contain devices or other storage groups. No mixing is permitted. This covers attempts to add devices to a parent storage group using `add`, `copy`, `move`, and `dg2sg`. This also covers attempts to add child storage groups to an storage group containing devices.
- A parent can have up to 32 child storage groups.
- Empty storage groups can be added to a parent storage group as long as the parent storage group inherits at least one device when the parent storage group is in a view.
- A parent storage group cannot inherit the same device from more than one child storage group.
- A child storage group may only be contained by a single parent storage group.
- No parent storage group can have a FAST association.
- A storage group already associated with a FAST policy is not allowed to be a parent storage group.
- Masking is not permitted for a child storage group which is contained by a parent storage group already part of a masking view.
- Masking is not permitted for the parent storage group which contains a child storage group that is already part of a masking view.
- A child storage group cannot be deleted until it is removed from its parent storage group.

Child storage group operation restrictions

The following restrictions exist for device operations involving child storage group device operations. This includes the `symsg add` and `addall` commands, as well as the `copy`, `copyall`, `move` and `moveall` commands because they involve adding devices to a storage group:

- A `moveall` operation is not permitted from a storage group that contains a masking view or is associated with a FAST policy.
- A `copy` or `copyall` operation is not permitted from a storage group that is associated with a FAST policy into a storage group that is associated with a FAST policy
- When in a view, the total number of devices inherited by a parent storage group cannot exceed 4096 devices.
- If adding Celerra devices into an storage group or a child storage group with Celerra devices to a parent storage group within a view, you must use the `-celerra` flag.
- If adding a CKD device or a child storage group with CKD devices to a parent storage group within a view, you must use the `-ckd` flag.
- If adding a RecoverPoint tagged device or a child storage group with RecoverPoint tagged devices to a parent storage group within a view, you must use the `-rp` flag.
- Adding an RecoverPoint tagged device to a storage group that is in a masking view containing FCoE directors is not allowed.
- Adding an AS400 device to a storage group that is in a masking view containing FCoE directors is not allowed.

The following restrictions exist for child storage group `remove` and `removeall` operations. This also includes `move` and `moveall` operations because they perform device removals:

- If the parent storage group has a masking view, any operations involving device removes from the child storage groups will not be permitted, if it causes the parent storage group to have no more devices. This includes removes and moves.

When they are performed on the parent storage group, the following `symsg` control operations affect all of the devices in all of the child storage groups contained by that parent:

- `ready` and `not_ready`
- `rw_enable` and `write_disable`
- `hold` and `unhold`
- `pin` and `unpin`

Removing devices from a storage group

Devices can be removed from a storage group singularly, using a combination of device ranges and single devices, or grouped in a text file.

To remove a single device from a storage group, use the following syntax:

```
symsg -sg SgName -sid SymmID [-i Interval] [-c Count][-V]
   remove dev SymDevName [-FORCE]
```

To remove multiple devices, or devices in a range, or a file, use the following syntax:

```
symsg -sg SgName -sid SymmID [-i Interval] [-c Count][-v]
   [-SA # | ALL] [-P #] [-N #]
   [-cap # [-captype mb | cyl]][-celerra] [-rp] [-ckd]
   [-devs <SymDevStart:SymDevEnd | SymDevName
   [,SymDevStart:SymDevEnd | SymDevName>...] >|
   -file DeviceFileName [-tgt]>]

 rmall [-force]
```

To remove all devices that are listed in a text file named `storgrp_a.txt` from a storage group named `prod` on array 123, enter:

```
symsg -sid 123 -file storgrp_a.txt -sg prod rmall
```

All storage devices within text file `storgrp.txt` are removed from storage group `prod`.

Device text files can be in either a one or two column format, specifying source and target devices. If the `-tgt` option is specified, only devices from the second column are removed from the storage group.

Note: As a restriction, you cannot remove all (`rmall`) devices in a storage group out of that group, if the group is part of a masking view or associated with a FAST policy.

Removing cascaded storage groups

You can remove child storage groups individually from a parent storage group using the `symsg remove sg` command. The following restrictions apply to `symsg remove sg` operations:

- If the parent storage group has a masking view, any operations involving device removes from the child storage groups will not be permitted, if it causes the parent storage group to have no more devices. This includes removes and moves.
- You cannot remove devices from a child storage group using the parent storage groups name.
- If you are removing Celerra devices from a storage group or a child storage group with Celerra devices from a parent storage group within a view, you must use the `-celerra` flag.
- If your are removing a CKD device or a child storage group with CKD devices from a parent storage group within a view, you must use the `-ckd` flag.
- If you are removing an RecoverPoint tagged device or a child storage group with RecoverPoint tagged devices from a parent storage group within a view, you must use the `-rp` flag.

The `symsg` command uses the following syntax for removing child storage groups from a parent storage group:

```
symsg -sg SgName -sid SymmID [-i Interval]
          [-c Count] [-v] [-celerra] [-rp] [-ckd]
        remove sg SgName1 [,SgName2,SgName3,SgNamen]
```

Setting Host I/O Limits for storage groups

Note: The Host I/O Limits feature requires Enginuity 5876 Q42012SR or HYPERMAX OS 5977.

The VMAX array Host I/O Limits feature allows you to define limits to enforce service levels and make application performance even more predictable. The Host I/O Limits settings (`-bw_max MBperSec` and `-iops_max IOperSec`) allow you to limit front-end (FE) port performance by setting FE bandwidth limits on a storage group. This feature is used to limit the amount of FE bandwidth and I/Os per second (IOPs) that can be consumed by a set of devices over a set of director ports. The bandwidth and I/Os controls are then monitored by the VMAX array to ensure that they do not exceed the specified maximum bandwidth or maximum IOPs. This feature allows you to place limits on the FE bandwidth and IOPs consumed by applications on the array.

Note: For additional documentation on using this feature, refer to the *Host I/O Limits for Symmetrix Family Arrays Technical Notes*, which are available on EMC Online support at: `https://support.EMC.com`. The paper explains the benefits of implementing the Host I/O Limits feature and provides use case examples.

Restrictions for storage groups with defined host I/O limits

The following restrictions apply to a storage group that has defined Host I/O Limits:

- At any given time, a storage group with defined Host I/O Limits can be associated with at most, one port group in any provisioning view. This means, that if the storage group with defined Host I/O Limits is in a provisioning view with a port group, the storage group and port group combination must be used when creating other provisioning views on this storage group. If you attempt to create the view using a different port group, the following error returns:

  ```
  The operation cannot be performed because the storage group with a
  Host I/O Limit can be associated with at most one port group in any
  masking view.
  ```

- Creating a provisioning view on child storage group is not allowed if the parent storage group has defined Host I/O Limits. If you attempt to create a view, the following error returns:

  ```
  The operation can not be performed because the child storage group
  or the parent storage group has a Host I/O Limit defined.
  ```

- Setting Host I/O Limits on a parent storage group is not allowed if any child storage group is already part of a provisioning view that was created using the child storage group. If you attempt to create a view, the following error returns:

  ```
  Cannot perform the requested operation because the group is currently
  within a masking view or a masking view through cascading.
  ```

- Any device can be in at most one storage group with Host I/O Limits. If you attempt to add the same device to another storage group with Host I/O Limits, the following error returns:

  ```
  The operation cannot be performed because the device already exists
  in a storage group with a Host I/O Limit.
  ```

- If a device is in a masking view with Host I/O Limits, it cannot be in another masking view without Host I/O Limits. If you attempt to add a device in a masking view with Host I/O Limits to another masking view without Host I/O Limits, the following error returns:

```
The operation cannot be performed because the device already exists
in a masking view with or without Host I/O Limit.
```

- If the Host I/O Limits (either `-bw_max` or `-iops_max`) being configured on a child storage group is greater than what is set on the parent storage group, then the operation will fail.
- If the Host I/O Limits (either `-bw_max` or `-iops_max`) on the child storage group that is being added is greater than what is configured on the parent storage group, then the operation will fail.
- If a Host I/O Limits (either `-bw_max` or `-iops_max`) being configured on a parent storage group is less than the what is set on any one of its child storage groups, then the operation will fail.
- The Host I/O Limits dynamic (`-dynamic`) setting for the parent and children must match. Child storage groups inherit the dynamic distribution setting from the parent storage group if set.

Setting the Host I/O Limit

The `symsg create` command provides two options for setting the storage group Host I/O Limit (`-bw_max` *MBperSec* and `-iops_max` *IOperSec*).

- The `-bw_max` option specifies the front-end maximum bandwidth in MBs/sec for the storage group. The valid range for bandwidth is from 1 MB/Sec to 100,000 MB/Sec.
- The `-iops_max` option specifies the front-end maximum IOs/sec. The valid range for IOPs is from 100 IO/Sec to 2,000,000 IO/Sec and must be specified in units of 100 IO/Sec. An error returns if an invalid range is specified.

To configure Host I/O Limits on a set of devices, the FE limits are added to a storage group. When you create a provisioning view using that storage group, the limits are applied to the devices in the storage group for the ports defined in the port group.

To create an empty storage group named prod with a FE bandwidth limit of 40,000 MB per second on array ID 123, enter:

```
symsg -sid 123 create prod -bw_max 40000
```

Changing the Host I/O limit

The `symsg set` command also provides two options for setting the storage group Host I/O Limit. Use the following syntax to set or change performance limits for an existing storage group:

```
symsg -sg SgName -sid SymmID [-i Interval] [-c Count]
   set [-bw_max MBperSec | NOLIMIT] [-iops_max IOperSec | NOLIMIT]
```

The same range limits described for the `create` command apply, but if `NOLIMIT` is specified, then the maximum bandwidth or IOs/Sec is set to unlimited.

To set an IOPs limit of 50,000 I/Os per second for a storage group named prod on array ID 123, enter:

```
symsg -sid 123 set prod -iops_max 50000
```

Host I/O Limits can be added, removed, or modified for a storage group. For Enginuity 5876Q22013SR and HYPERMAX OS 5977, the Host I/O Limit for a cascaded storage group can be added for both the parent and the child storage group. If a parent storage group

has a control set, the setting is shared among all its child storage groups when a provisioning view is created using the parent storage group. If a parent storage group has a control set, you cannot create provisioning views using the child storage groups.

If the Host I/O Limits (either `-bw_max` or `-iops_max`) being configured on a child storage group is greater than what is set on the parent storage group, then the operation will fail. If a Host I/O Limits (either `-bw_max` or `-iops_max`) being configured on a parent storage group is less than the what is set on any one of its child storage groups, then the operation will fail.

Note: With Enginuity 5876Q22013SR and HYPERMAX OS 5977, the `symsg set`command allows Host I/O Limits to be set on a storage group that is in a provisioning view using a port group that has FCoE ports.

"Listing storage groups" on page 109 provides information on listing storage group information including Host I/O Limit and demand reports.

Dynamic distribution setting for FE Host I/O limits

With Enginuity 5876Q22013SR and HYPERMAX OS 5977, you can optionally configure a dynamic distribution of the FE Host I/O limits by using the `-dynamic` option. Setting port failure capability causes the fraction of the configured Host I/O limits available to a configured port to be adjusted, based on the number of ports that are currently online. Setting dynamic distribution causes the configured limits to be dynamically distributed across the configured ports, allowing the limits on each individual port to adjust to fluctuating demand. The Dynamic Distribution feature is supported on Front-End ports from Fibre Channel, iSCSI and FCoE directors.

If port failure capability is set on the FE Host I/O Limit and one of the directors is taken offline or fails, its percentage of the limit is then redistributed, and the two remaining ports can consume the remaining portion of the limit. Once the faulted director comes back online, the limit is again redistributed across all three ports.

If dynamic distribution is set on the FE Host I/O Limit, then the maximum IOPs for the provisioned view are distributed dynamically across all three of the active ports within the view. The distribution fluctuates based on the current demand on each of the ports

The `-dynamic` option is available for the `symsg set` and `create` commands to allow setting of the dynamic distribution attribute type on a storage group. If the option field is not given a default of `NEVER` is used.

When the dynamic distribution attribute is set on a parent storage group, every child storage group inherits the same attribute value. The operation is blocked If you attempt to set the attribute on a child storage group.

Note: The parent and child storage groups must have the same dynamic distribution attribute value before they can be placed in a cascaded relationship.

Use the following syntax to set or change dynamic distribution for an existing storage group:

```
symsg -sid <SymmID> [-i <Interval>] [-c <Count>] [-v]

   create <SgName> [-bw_max <MBperSec>]
                   [-iops_max <IOperSec>]
```

```
                                [-dynamic <NEVER | ALWAYS | ONFAILURE>]
          . . .
     symsg -sg <SgName> -sid <SymmID> [-i <Interval>]
          [-c <Count>]

        set <[-bw_max <<MBperSec> | NOLIMIT>]
             [-iops_max <<IOperSec> | NOLIMIT>]
             [-dynamic <NEVER | ALWAYS | ONFAILURE>]>
```

Setting the `-dynamic` option to `ALWAYS` indicates that the Host IO Limits for the storage group should always be dynamically redistributed. Setting to `NEVER` indicates that the Host IO Limits for the storage group should never be dynamically redistributed and setting to `ONFAILURE` indicates that the Host IO Limits for the storage group should dynamically redistributed only upon afailure of a Front-End Port.

Note: Use the `symsg show` command to display the value of a storage group's Dynamic Distribution setting. If a provisioning view has not been created on the parent storage group, the Dynamic Distribution displays as N/A.

Modifying the host ID of devices in a storage group (HYPERMAX OS 5977)

Arrays running HYPERMAX OS 5977 are assigned a new device host ID format, a Federated ID that is globally unique and transportable between storage systems. For devices, you can choose a host ID type that is either the new Federated ID format (`native`) or an ID compatible with the previous ID format (`compatible`) that is a non-portable ID value only unique within the array.

By using the `set -host_id` option, you can modify the host ID of devices in a storage group:

```
symsg -sg <SgName> -sid <SymmID> [-i <Interval>]
          [-c <Count>] [-v] [-noprompt] [-rp] [-star]
          [-celerra] [-skip -lockid <lockNum>]

        ready

        not_ready
         . . .

        set -host_id <compatible | native>
```

Restrictions

- When `compatible` is specified, if the device number of any of the devices in the storage group is greater than 0xFFFF, the operation will fail.
- The operation will fail if any of the devices are online.
- The operation will fail if any of the devices are visible to a host through either mapping or masking.
- The operation will fail if `native` is specified and the emulation type of the device is AS/400_D910_099 on an array running HYPERMAX OS 5977.

Copying devices from a storage group to a device group

Devices from an existing storage group can be copied (or added) to a device group. The copied devices remain in the storage group (*SgName*) and are added to the destination device group (*DgName*).

Use the following syntax to add devices from an existing storage group to a device group:

```
symsg -sid SymmID [-i Interval] [-c Count] [-v]
      sg2dg SgName DgName [-bcv | -vdev | -tgt]
      [-R1 | -R2 | -R21 | -noRDF]
```

Adds RDF1 (-R1) devices, RDF2 (-R2) devices, RDF21 (-R21) devices, or non-SRDF devices (-noRDF) to the device group. This option must match the device type.

If the device group does not exist, an error is returned.

To copy only R1 devices belonging to group number 008 from storage group `prod to device group prod_2`, enter:

```
symsg sg2dg prod prod_2 -R1 -sel_rdfg 008
```

Note: For cascaded storage groups, the `sg2dg` command behavior has been modified to create a device group from a parent storage group with the device group containing devices from all the child storage groups.

Listing storage groups

To retrieve a list of all storage groups defined in your array, enter:

```
symsg list
```

The `symsg list` command returns a list of all storage group names, the number of devices and gatekeepers each contains, and indicates whether the storage group is associated with a FAST policy, contained in a masking view, is the parent or child in a cascaded relationship, if a Host I/O Limit is defined, shared, or both. Note that for parent storage groups, the display includes both the number of child storage groups it contains and the cumulative total of all devices contained by those child storage groups.

The following example displays all storage groups for array 000197100230:

```
symsg -sid 230 list

        S T O R A G E  G R O U P S


Symmetrix ID:        000197100230

                       Flags Number  Number Child
Storage Group Name     EFMSL Devices GKs    SGs
--------------------------------------------------
AcctPay                FX...     250     13     0
Customer               AX...      38      2     0
Ordering               MX...      19      1     0
Payroll                FX...      28      1     0
TestOrdering           .X...      10      0     0
Testsg                 .....      10      0     0

Legend:
  Flags:
    Device (E)mulation  A = AS400, F = FBA, 8 = CKD3380,
```

```
                                9 = CKD3390, M = Mixed, . = N/A
     (F)ast                     X = Fast Managed, . = N/A
     (M)asking View             X = Contained in Mask View(s),   . = N/A
     Cascade (S)tatus           P = Parent SG, C = Child  SG,    . = N/A
     Host IO (L)imit            D = Defined, S = Shared, B = Both, . = N/A
```

Note: For the (M)asking View flag, if a Parent storage group contains a masking view, all its child storage groups show that they are contained within the same masking view.

Note: Storage groups with names longer than 21 characters display with their first 21 characters followed by an asterisk (*). "Setting group names to display full names" on page 31 explains how to set the environmental variable to display full group names.

Host I/O limit demand reports

You can specify Host I/O Limit demand options with the `symsg list` command to report the Host I/O Limit demand on each individual director port (`-by_port -demand`) or port group (`-by_pg -demand`). The demand reports show that the Host I/O Limit is divided equally among all of the directors in the port group, independent of the number of ports on each director. This means that the demand reports show the same Host I/O Limit on both ports even though the Host I/O Limit is shared by the ports. Because of this, it is recommended that you configure only one of the ports of a director in the same port group.

Note: Enginuity 5876Q42012SR or HYPERMAX OS 5977 is required for this feature.

On arrays running Enginuity 5876, all front-end emulations support a fixed number (up to two) of ports. Multiple front-end emulations can exist on the same director board, providing additional host connectivity, but all front-end directors are limited to one or two physical ports.

On arrays running HYPERMAX OS 5977, only a single front-end emulation of each type (FA, EF, etc.) can be assigned to each director, however each of these emulations can be assigned a variable number of physical ports (up to 32, numbered from 0 - 31). In addition, directors containing a Fibre Channel emulation have 32 virtual ports (numbered 32 - 63), that are reserved for use by internal guests.

To list Host I/O Limit demands, use the following syntax:

```
symsg -sid SymmID [-i Interval] [-c Count] [-V]
   list -by_port -demand [-pg PgName | -dir <# [-p # | ALL] | ALL>]
   list -by_pg -demand [-pg PgName]
```

To list a demand report for all director ports, enter:

```
symsg list -by_port -demand


Symmetrix ID   : 000197100001

 Director       IO Limit             Bandwidth Limit
-------------- ---------------- --------------------------------------
               Maximum  Number     Port    Maximum    Number
         Flags   Demand Nolimit    Speed     Demand   NoLimit   Excess
DIR:Port  HD   (IO/Sec)     SGs (MB/Sec) (MB/Sec) (%)     SGs (MB/Sec)
```

```
-------- ----- -------- ------- -------- -------- --- ------- --------
01A:001   NN          0       0     1000        0   0       0    +1000
01A:010   YN       2000       0     1000     1000 100       0       +0
  . . .
Legend:
  Flags:
    (H)ost I/O Limit Exists     Y = Yes, N = No, M = Mixed, . = N/A
    (D)ynamic Distibution       Y = Yes, N = No, . = N/A
  . . .
```

- The Flags H (Host I/O Limit Exist) column indicates whether a limit demand has been configured on a port, and if storage groups with no limits configured are sharing the port. The Flags D (Dynamic Distribution) column indicates if the dynamic option is set.
- The Maximum Demand columns indicate the total Host I/O Limit demand on the specified director port in MB/Sec or IO/Sec.
- The Port Speed column indicates the bandwidth in MB/Sec for that port (the port negotiated speed).
- The Number Nolimit SGs columns indicate the number of storage groups with no limits configured that are sharing the port.
- The Excess column indicates the amount of bandwidth in MB/sec that is available on the director port after accounting for the demands on the port.

Note: Only storage groups that have a provisioning view created on them are shown as placing a Host I/O Limit demand on the director port. If both the parent and child storage group has a Host I/O defined, the maximum demand is calculated based on the parent Host I/O settings.

To list a demand report for all port groups, enter:

```
symsg list -by_pg -demand

Symmetrix ID   : 000195700123

Port Group                IO Limit             Bandwidth Limit
----------------------- ---------------- ---------------------------------------
                  Host   Maximum  Number Port Grp    Maximum    Number
                 Limit  Demand  Nolimit    Speed     Demand   NoLimit   Excess
Name              Exist (IO/Sec)     SGs (MB/Sec) (MB/Sec) (%)     SGs (MB/Sec)
----------------- ----- -------- ------- -------- -------- --- ------- --------
PG_Eng1           Yes          0       1     2000     3000 150       0    -1000
PG_Eng2           Yes       1500       0     2000     3000 150       0    -1000
PG_Eng3           Yes          0       1     1000        0   0       1    +1000
PG_Eng4           No        2000       0     1000     1000 100       0        0
  . . .
```

Note: Only storage groups that have a provisioning view created on them will be shown as placing a Host I/O Limit demand on the port group.

- The Host Limit Exist column indicates whether a limit demand has been placed on a port group, and if storage groups with no limits configured are sharing the port group.
- The Maximum Demand columns indicate the total Host I/O Limit demand on the specified port group in MB/Sec or IO/Sec.
- The Port Grp Speed column indicates the bandwidth in MB/Sec for that port group (the aggregated port negotiated speed for the ports in the group).

- The Number Nolimit SGs columns indicate the number of storage groups with no limits configured that are sharing the port group.
- The Excess column indicates the amount of bandwidth in MB/sec that is left available on the port group after the demands have been accounted for.

The following example shows a verbose (`-v`) demand report for all director ports. The verbose option can also be used for port groups.

```
symsg list -by_port -demand -v

Symmetrix ID   : 000197100001

Director Identification            : FA-1A
Director Port                      : 001
WWN Port Name                      : 5000097300092150
Host I/O Limit Exists              : No
Dynamic Distribution               : No
Port Total Demand (IO/Sec)         :        0
Number of SGs without Limit (IO/Sec):        0

Port Negotiated Speed (MB/Sec)     :     1000
Port Total Demand (MB/Sec)         :        0
Percent Port Capability (%)        :        0
Port Excess (MB/Sec)               :    +1000
Number of SGs without Limit (MB/Sec):        0


Director Identification            : FA-1A
Director Port                      : 010
Port Total Demand (IO/Sec)         :     2000
Number of SGs without Limit (IO/Sec):        0

Port Negotiated Speed (MB/Sec)     :     1000
Port Total Demand (MB/Sec)         :     1000
Percent Port Capability (%)        :      100
Port Excess        (MB/Sec)        :        0
Number of SGs without Limit (MB/Sec):        0

Storage Groups (1)
       {
       ---------------------------------------------
                                   Maximum  Demand
```

Showing details about a storage group

Use the `symsg show` command to gather detailed information about a specific storage group. For example, to show information about storage group `SG_Corp1` on array 000195700601, enter:

```
symsg show -sid 601 SG_Corp1
```

Note: The example output below is truncated and is intended to provide the overall example of a possible data set.

```
symsg show SG_Corp1

Name: SG_Corp1

   Symmetrix ID             : 000195700601
   Last updated at          : Wed Apr 09 00:28:16 2012
   Masking Views            : Yes
   FAST Policy              : Yes
```

```
   Host I/O Limit           : None
   Host I/O Limit MB/Sec    : N/A
   Host I/O Limit IO/Sec    : N/A
   Dynamic Distribution     : N/A
   Number of Storage Groups : 0
   Storage Group Names      : N/A

   Devices (10):
    {
    ----------------------------------------------------------
    Sym                                Device                Cap
    Dev    Pdev Name                   Config        Sts     (MB)
    ----------------------------------------------------------
    0A41   N/A                         2-Way Mir      RW     2000
    0A42   N/A                         2-Way Mir      RW     2000
      . . .
```

Note: The above example shows a storage group without a defined Host I/O Limit.

The `symsg show` command can be used to display parent and child information for cascaded storage groups. Using the `symsg show` command on a parent storage group displays devices from all child storage groups in ascending sorted order.

The following example shows information for a parent storage group:

```
symsg show SG_Eng1

Name: SG_Eng1

   Symmetrix ID             : 000195700601
   Last updated at          : Wed Apr 11 00:28:16 2012
   Masking Views            : Yes
   FAST Policy              : No
   Host I/O Limit           : Defined
   Host I/O Limit MB/Sec    : 1000
   Host I/O Limit IO/Sec    : NoLimit
   Dynamic Distribution     : N/A
   Number of Storage Groups : 2
   Storage Group Names      : Eng1_Data       (IsChild)
                              Eng2_Data       (IsChild)
   Number of Composite Groups : 0
   Composite Group Names      : N/A
   Number of Groups           : 2
   Group Names                : DG1
                              : DG2
   Devices (20):
    {
    ----------------------------------------------------------
    Sym                                Device                Cap
    Dev    Pdev Name                   Config        Sts     (MB)
    ----------------------------------------------------------
    1A41   N/A                         2-Way Mir      RW     2000
    1A42   N/A                         2-Way Mir      RW     2000
    1A43   N/A                         2-Way Mir      RW     2000
    1A44   N/A                         2-Way Mir      RW     2000
    1A45   N/A                         2-Way Mir      RW     2000
      . . .
```

Note: The above example shows a parent storage group with a defined Host I/O Limit.

The following example shows information for a child storage group:

```
symsg show Eng1_Data

Name: Eng1_Data
```

```
Symmetrix ID             : 000195700601
Last updated at          : Wed Apr 11 00:28:16 2012
Masking Views            : Yes
FAST Policy              : No
Host I/O Limit           : Shared
Host I/O Limit MB/Sec    : 1000
Host I/O Limit IO/Sec    : NoLimit
Dynamic Distribution     : N/A
Number of Storage Groups : 1
Storage Group Names      : SG_Eng1        (IsParent)

Devices (10):
 {
 ----------------------------------------------------------
 Sym                              Device               Cap
 Dev    Pdev Name                 Config        Sts    (MB)
 ----------------------------------------------------------
 1A41   N/A                       2-Way Mir      RW    2000
 1A42   N/A                       2-Way Mir      RW    2000
 1A43   N/A                       2-Way Mir      RW    2000
 1A44   N/A                       2-Way Mir      RW    2000
 1A45   N/A                       2-Way Mir      RW    2000
   . . .
```

Note: The above example shows a child storage group when it is sharing the Host I/O Limit defined in the parent storage group. This is only shown when a provisioning view is created using the parent storage group. If a provisioning view was not created on the parent storage group, the limit is not reported as being shared.

Note: Host I/O Limit fields are only displayed for Solutions Enabler V7.5 with Enginuity 5876Q42012SR or higher.

For a storage group that is not part of a cascaded relationship, the following fields display:

```
Number of Storage Groups          : 0
Storage Group Names               : N/A
```

Exporting and importing device lists

You can save the list of devices from an existing storage group to a text file on your host system, and this file can later be imported to create a storage group. This can be useful if you then delete the group and later wish to recreate it. The device list file contains a device description line for as many devices as there are listed in the storage group. Lines that are blank or include a pound (#) sign in the first column are ignored.

Group files for the `import` and `export` commands contain device parameters in the following format:

```
SymmID
SymDevName
SymDevName
. . .
```

Group files for the `importall` and `exportall` commands contain device parameters in the following format:

```
SgName
SymmID
SymDevName
SymDevName
```

. . .
This format can be repeated in the file for multiple storage groups.

Note: If a filename is not specified with the `symsg import` or export command, a default file named `symsg.txt` is created in the directory where the `symsg` command is executed. If a filename is not specified with the `symsg importall` or `exportall` command, a default file named `symsgall.txt` is created in the directory where the `symsg` command is executed.

Exporting a device list

You can export a list of devices in a storage group to a file on your host system using the following syntax:

```
symsg -sid SymmID [-i Interval] [-c Count][-v]

   export SgName [-f FileName]
   exportall [-f FileName]
```

For example, to export the storage group membership from group `prod2` to device file named `prod2list.txt`, enter:

```
symsg -sid 123 export prod2 -f prod2list.txt
```

The `symsg export/exportall` command output includes an identifier (S), to denote that the line contains the name of a child storage group.

For example, the output of an export of parent storage group `sgparent` displays as follows:

```
000194900341
S sgchild1
S sgchild2
```

An example of the output of an exportall is:

```
<sgparent>
000194900341
S sgchild1
S sgchild2

<sgchild1>
000194900341
00201
00206

<sgchild2>
000194900341
00404
00405
00406
```

For arrays running HYPERMAX OS 5977, `symsg export/exportall` supports exporting the Service Level Objective, the workload, the Storage Resource Pool and the RDF coordination set on the storage group. If a storage group has a Service Level Objective or a Storage Resource Pool name set on the storage group, the name is copied to the export file. If an implicitly set Optimized Service Level Objective or the system default Storage Resource Pool for the emulation type is set on the storage group, the string `DEFAULT` is written in place of the name.

Importing a device list

You can add multiple devices to a new or existing storage group by importing an existing file that contains a list of devices created by the export action using the following syntax:

```
symsg -sid SymmID [-i Interval] [-c Count][-v]

   import SgName [-f FileName]
   importall [-f FileName]
```

For example, to create a storage group named `prod2` from the device file `prod2list.txt`, enter:

```
symsg -sid 123 import prod2 -f prod2list.txt
```

Typically, you import files that were previously exported (refer to "Exporting a device list" on page 86).

The `import` action will create the storage group if the group name specified in the command does not already exist, or you can import to an existing group name that is partially populated. If you import to an existing group, the devices in the imported file will be appended to the existing group membership.

In addition, you can recreate all storage groups, using the `symsg importall` command, that were from data contained in a text file previously created using the `symsg exportall` command.

Note: For `importall`, if any of the storage groups that you you are attempting to create already exist, Solutions Enabler displays a message indicating its existence, and the operation will continue to create the next storage group in the list.

The following security privileges are required to execute this command:

- Required Access type: BASE
- Required Authorization Rights: Storage Admin.

HYPERMAX OS 5977

For VMAX arrays running HYPERMAX OS 5977, `symsg import/importall` supports importing devices from the FAST-managed storage group in the import file to a target storage group.

- If the imported storage group exists in the array, the target storage group will have its Service Level Objective, workload, Storage Resource Pool, and RDF Coordination overwritten with those from the imported storage group. If the imported storage group was FAST-managed, the resulting target storage group will become FAST-managed. If the imported storage group was not FAST-managed, the resulting target storage group will become non FAST-managed.
- If the Service Level Objective set on the storage group does not exist on the target array, the storage group will be imported and set to use an Optimized Service Level Objective with no workload.
- If the workload set on the storage group does not exist on the target array, the storage group will be imported without a workload.

- If the Storage Resource Pool set on the storage group does not exist on the target array, the storage group will be imported and set to the default Storage Resource Pool for the emulation type.
- If the both Service Level Objective and the Storage Resource Pool set on the storage group do not exist on the target array, the storage group will be restored with no Service Level Objective and no Storage Resource Pool and will not be FAST-managed.
- If the resulting target storage group is FAST-managed, the command will fail if any device in the storage group is already in another FAST-managed storage group on the array or if any device in the storage group is an encapsulated device.
- The command will fail if the Service Level Objective used by the storage group cannot be supported based on the Storage Resource Pool that is being used by the storage group.

Enginuity versions 5773 through 5876

If the target is an array running Enginuity 5876 or lower, the Service Level Objective name, workload, Storage Resource Pool name, and RDF coordination will not be copied.

Renaming, moving, copying, and deleting storage groups

The `symsg` command also provides the ability to rename, move, copy, and delete existing storage groups.

Note: A parent storage group may not be specified as the destination for a `move`, `moveall`, `copy` or `copyall` operation, as a parent storage group may not contain both child storage groups and devices.

Renaming a storage group

To rename an existing storage group, use the following syntax:

```
symsg -sid SymmID [-i Interval] [-c Count]
   rename OldSgName NewSgName
```

To rename a storage group named `prod` on array ID#59866000123 to `prod_B`, enter:

```
symsg rename -sid 123 prod prod_B
```

Moving a device

Use the `symsg move dev` command to move one device, specifying the device name, from one existing storage group to another existing storage group. The moved device is deleted from the current storage group (*SgName*) and added to the destination storage group (*DestSgName*).You can specify the interval and count options (`-i` and `-c`) to wait a predetermined time (interval) between attempts (count) to acquire an exclusive lock on the host database.

Starting with Solutions Enabler V8.0.1, the `-force` flag is no longer required for non-disruptive move operations. Moving a device to another storage group will not disrupt the host visibility for the device if any of the following conditions is met:

- Moving devices between child storage groups of a parent storage group when the view is on the parent storage group.

- Moving devices between storage groups when a view is on each storage group and both the initiator group and the port group are common to the views.
- Moving devices between storage groups when a view is on each storage group and they have a common initiator group. They have different port groups but the same set of ports.
- Moving devices between storage groups when a view is on each storage group and they have a common initiator group. They have different port groups but the target port group is a superset of the source port group (additional ports).

  Note that iIf the initiator group has the consistent LUN flag set and Solutions Enabler cannot assign consistent LUNs on the additional ports, the operation will fail.
- The source storage group is not in a masking view.

If none of the conditions are met, the operation is rejected but the move can be forced by specifying the `-force` flag. Note that forcing a move may affect the host visibility of the device.

Use the following syntax to move an individual device:

```
symsg -sg <SgName> -sid <SymmID> [-i <Interval>]
          [-c <Count>] [-v] [-celerra] [-rp] [-ckd]

. . .

        move dev <SymDevName> <DestSgName> [-force]
```

For example, to move device 30 on array ID#59866000123 from storage group `prod` to storage group `test`, enter:

```
symsg -sid 123 -sg prod move dev 30 test
```

Restrictions

HYPERMAX OS 5977

- You can move the selected or all the devices in a source storage group to a standalone or child destination storage group that is FAST-managed. The operation will fail if after the move the device is in more than one FAST-managed storage group or if moving encapsulated devices to the storage group.

Enginuity versions 5773 through 5876

- You cannot move the last device in a storage group out of the group, if the group is part of a masking view or associated with a FAST policy.
- If the devices are associated with a FAST policy, the `symsg move` and `moveall` commands do not work to move devices between storage groups. You must first remove the devices from the original storage group using the `symsg remove` or `rmall` command, and then add them to the new storage group using `symsg add`.
- If the parent storage group has a masking view, any operations involving device removes from the child storage groups are not permitted if the operation causes the parent storage group to have no more devices. This includes removes and moves.

Moving all devices

Multiple devices can be moved from a storage group in a list, using a combination of device ranges and single devices, or grouped in a text file. The moved devices are deleted from the current storage group (*SgName*) and added to the destination storage group (*DestSgName*). To move multiple devices, devices in a range, or listed a file, use the following syntax:

```
symsg -sg SgName -sid SymmID [-i Interval] [-c Count][-v]
   [-SA # | ALL] [-P #] [-N #]
   [-cap # [-captype mb | cyl]][-celerra] [-rp] [-ckd]
   [-devs <SymDevStart:SymDevEnd | SymDevName>
   [<,SymDevStart:SymDevEnd | SymDevName...]> |
   -file DeviceFileName -tgt]

   moveall DestSgName
```

When choosing to move all devices from one existing storage group to another, you can move all devices, or use the additional options to select specific devices. Specify the interval and count options (`-i` and `-c`) to wait a predetermined time (interval) between attempts (count) to acquire an exclusive lock on the host database.

Restrictions

HYPERMAX OS 5977

- You can move the selected or all the devices in a source storage group to a standalone or child destination storage group that is FAST-managed. The operation will fail if after the move the device is in more than one FAST-managed storage group or if moving encapsulated devices to the storage group.

Enginuity versions 5773 through 5876

- You cannot move all devices out of a storage group if the group is part of a masking view or associated with a FAST policy.

For example, to move all devices from a storage group named `prod` to a storage group named `test` on array 123, enter:

```
symsg -sid 123 -sg prod moveall test
```

To move multiple devices from a storage group named `prod` to a storage group named `test` on array 123, enter:

```
symsg -sid 123 -sg prod moveall -devs 31:35,37,40:43
```

Storage devices 31 through 35, 37, and 40 through 43 are moved from storage group `prod` to storage group `test`.

The `symsg moveall` command requires the `-force` option under the following conditions:

- Moving devices from a source storage group contained in a masking view, either directly or indirectly (inherited from a parent storage group contained in a masking view) to a destination storage group that is also contained in a masking view, either directly or indirectly.
- Moving devices between two storage groups, both associated with a FAST policy.

Copying a device

Use the `symsg copy dev` command to copy one device, specifying the array device name, from one existing storage group to another existing storage group. The copied device remains in the current storage group (*SgName*) and is added to the destination storage group (*DestSgName*). You can specify the interval and count options (`-i` and `-c`) to wait a predetermined time (interval) between attempts (count) to acquire an exclusive lock on the array host database. The following syntax is used to copy an individual device:

```
symsg -sg SgName -sid SymmID [-i Interval] [-c Count][-v]
copy dev SymdevName DestSgName
```

For example, to copy device 30 on array ID#59866000123 in device group `prod` to device group `test`, enter:

```
symsg -sid 123 -sg prod copy dev 30 test
```

Device 30 is now contained in both storage group `prod` and storage group `test`.

Restrictions

HYPERMAX OS 5977

- You can copy the selected or all the devices in a source storage group to a standalone or child destination storage group that is FAST-managed. However, the command will fail if the device is already in another storage group that is FAST-managed or if copying encapsulated devices.

Enginuity versions 5773 through 5876

- You cannot copy a device from a storage group that is associated with a FAST policy into another storage group that is also associated with a FAST policy.

Copying all devices

Multiple devices can be copied from a storage group in a list, range, or grouped in a text file. The copied devices remain in the current storage group (*SgName*) and are added to the destination storage group (*DestSgName*).

Use the following syntax to copy multiple devices from a storage group:

```
symsg -sg SgName -sid SymmID [-i Interval] [-c Count][-v]
    [-SA # | ALL] [-P #] [-N #]
    [-cap # [-captype mb | cyl]][-celerra] [-rp] [-ckd]
    [-devs <SymDevStart:SymDevEnd | SymDevName >
    [<,SymDevStart:SymDevEnd | SymDevName...]> |
    -file DeviceFileName [-tgt]]

    copyall DestSgName
```

When choosing to copy all devices from one existing storage group to another, you can copy all devices, or use the additional options to select specific devices. Specify the interval and count options (`-i` and `-c`) to wait a predetermined time (interval) between attempts (count) to acquire an exclusive lock on the host database.

Restrictions

HYPERMAX OS 5977

- You can copy the selected or all the devices in a source storage group to a standalone or child destination storage group that is FAST-managed. However, the command will fail if the device is already in another storage group that is FAST-managed or if copying encapsulated devices.

Enginuity versions 5773 through 5876

- You cannot copy a device from a storage group that is associated with a FAST policy into another storage group that is also associated with a FAST policy.

For example, to copy all devices from a storage group named `prod` to a storage group named `test` on Smmetrix array 123, enter:

```
symsg -sid 123 -sg prod copyall test
```

All devices from storage group `prod` are now also contained in storage group `test`.

To copy multiple devices from a storage group named `prod` to a storage group named `test` on array 123, enter:

```
symsg -sid 123 -sg prod copyall -devs 31:35,37,40:43
```

Storage devices 31 through 35, 37, and 40 through 43 from storage group `prod` are now also contained in storage group `test`.

To copy multiple devices from a device file named `test_2.txt` to a storage group named test and a destination storage group (*DestSgName*) named `test_2` on array 123, enter:

```
symsg -sid 123 -sg test copyall -file test_2.txt test_2
```

All devices listed in device file `test_2.txt` are now contained in both storage group `test` and `test_2`.

Copying devices from a storage group to a device group

Devices from an existing storage group can be copied (or added) to a device group. The copied devices remain in the storage group (*SgName*) and are added to the destination device group (*DgName*). Use the following syntax to add devices from an existing storage group to a device group:

```
symsg -sid SymmID [-i Interval] [-c Count][-v]
sg2dg SgName DgName [-bcv | -vdev | -tgt]
```

If the device group does not exist, it will be created. If optional device types are not specified, only standard devices will be added.

To add devices from a storage group named `prod` on array 123 to a device group named `prod_2`, enter:

```
symsg -sid 123 sg2dg prod prod_2
```

All standard devices in storage group `prod` have been added to device group `prod_2`.

To add only the target devices from a storage group named `prod` on array 123 to a device group named `prod_2`, enter:

```
symsg -sid 123 sg2dg prod prod_2 -tgt
```

All target devices in storage group `prod` have been added to device group `prod_2`.

Note: An error will be returned if the `-tgt` option is specified and the storage group contains both standard and BCV devices.

Deleting a storage group

Use the `symsg delete` command to delete a storage group using the following syntax:

```
symsg -sid SymmID [-i Interval] [-c Count]
delete SgName [-force]
```

To delete an empty storage group named `prod` on array 123, enter:

```
symsg delete -sid 123 prod
```

Note: As a restriction, you cannot delete a storage group if the group is part of a masking view or associated with a FAST policy.

The following restrictions apply to the `symsg delete` command for cascaded storage groups:

- A parent storage group cannot be deleted unless the `-force` flag is used.
- A parent storage group cannot be deleted if it has a masking view
- A child storage group must be removed from its parent storage group before it can be deleted.

Performing miscellaneous operations on storage group devices

Use the following command syntax to perform the following operations on devices within a storage group:

- Change the state (write disable, read write enable, ready, not ready) of a device
- Set or reset the hold bit on a device
- Compress or uncompress a thin device
- Bind, unbind, rebind, start or stop allocations, free, and reclaim thin devices

Note: For Enginuity versions 5876 or lower, free and reclaim actions are blocked for any device which is the source or target of a Clone, Emulation, or Snap session.

- Set or unset persistence on a device
- Set the Optimized Read Miss mode
- Sets or unsets Geometry Compatible Mode (GCM)

  Note: For arrays running HYPERMAX 5977, the Optimized Read Miss feature is not supported.

```
symsg -sg sgName -sid SymmID [-i Interval]
          [-c Count] [-v] [-noprompt] [-rp] [-star]
          [-celerra] [-skip -lockid locknum]
```

```
not_ready

ready

rw_enable

write_disable

hold

unhold [-symforce]

compress [-stop]

uncompress [-stop]

bind -pool <PoolName>

rebind -pool <PoolName>

unbind

allocate [-persistent | -stop]

free [-stop]

reclaim [-persistent | -stop]

set -persistent

unset -persistent

set -orm [on | off | system]

set -gcm [-symforce]

unset -gcm [-symforce]
```

Note: "Showing details about thin devices" on page 465 provides information on compressing and uncompressing thin devices.

For more information on actions for thin devices (bind, unbind, rebind, allocate, etc.) refer to "Performing actions on thin devices (Enginuity 5773 through 5876)" on page 284.

For arrays running Enginuity 5876 Q22013SR and higher, the `symsg` command allows you to set the Optimized Read Miss mode to Off or System Managed for all devices in a storage group. "Optimized Read Miss mode (Enginuity 5876 only)" on page 439 provides additional information on this feature.

Composite groups

A composite group (CG) is a user-defined group comprised of devices or device groups, that can belong to one or more locally-attached arrays and one or more SRDF groups within a array. Members can be individual devices or device groups spanning multiple arrays and SRDF groups.

For example, with a composite group you can make a consistent, restartable local copy of a database that is using volumes spanning two local arrays. Without composite groups, it would be impossible to guarantee that all of the BCVs would be split at the same point in time.

Composite groups are created and managed by using the `symcg` command.

Device members

A single composite group can contain devices from the following different device lists:

- Standard device list (STD) — Non-BCV devices that are local to the host.
- Local Business Continuance Volume (BCV) list — BCV devices local to the host.
- Local VDEV list (VDEV) — Virtual devices that are local to the host.
- Remote VDEV list (RVDEV) — Virtual devices that are remote.
- Remote BCV list (RBCV) — BCV devices that are to be associated with the remote mirrors of the STD devices.
- BCV-Remote BCV list (BRBCV) — BCV devices that are to be associated with the remote mirrors of the local BCV devices.
- Remote-Remote BCV List (RRBCV) — Remote BCV devices that are to be associated with the remote mirrors of the RBCV devices.
- Local TGT list (TGT) — TF/Clone target devices that are local to the host.
- Remote TGT list (RTGT) — TF/Clone target devices that are remote.
- R21 STD devices — R21 devices utilize two mirrors and are considered to be concurrent SRDF devices.

Logical device name support

A composite group can contain logical device names (aliases) for devices. All new composite groups, and any devices added to them, will automatically be assigned an *LdevName* if you do not specify one.

The devices added to device groups and composite groups are assigned a default logical names at add time. The name is unique within the group to which it was added.

Device group members

A device group can be a member of more than one composite group.

The integrity of control operations is maintained separately at the device group and composite group level. For example, pairing STD devices with BCV devices is only allowed if both devices are contained within the same device group.

Device group membership of composite groups provides the following:

- Building blocks to creating composite groups.
- Support for auto-correct of the composite group.
- SRDF consistency at the composite group and SRDF group name levels.
- GNS support of the composite group.

Restrictions

Before you create a composite group containing device groups, review the following restrictions:

- GNS does not remotely mirror composite groups containing device groups.
- Composite groups can contain either individual devices or device groups, but not both.
- If any of the device groups contained by a composite group become invalid, the composite group also becomes invalid.
- Cannot add a device group to an enabled composite group.
- A composite group cannot contain multiple device groups that all have the same devices.
- Control operations are only allowed at the device group or composite group level if currently supported at that level. For example, control operations such as `symdg` and `symqos` are allowed at the device group level.
- A device group must exist before it can be added to the composite group.
- The device groups in a composite group must follow the guidelines and restrictions outlined in "Device lists" on page 79 and "Creating a device group" on page 80.

Adding device groups to a composite group

When adding more than one device group to a composite group, a device group may contain devices with the same logical names, resulting in a composite group having more than one device with the same logical name. To prevent this, the logical name of a device in a device group is not carried over into the composite group. A new logical name is created for each device using the following format:

xxxxxsssss_ddddd

where:

- xxxxx is the one of the following reserved words representing the device type:
  - DEV
  - BCV
  - RBCV
  - BRBCV
  - RRBCV
  - 2BCV

- VDEV
- RVDEV
- 2VDEV
- TGT
- RTGT
- 2TGT

◆ `sssss` is the last five characters of the array ID containing the device.

◆ `ddddd` is the device number, in hexadecimal, which is guaranteed to be unique within the array.

These naming conventions guarantee that all devices have unique logical names from the composite group point of view. Devices in the device groups maintain the logical names that were assigned at the time when the devices were added to the composite group.

Composite group types

You create a composite group using one of these types:

◆ REGULAR (Solutions Enabler V7.4 and higher allows the inclusion of SRDF devices)

◆ RDF1 (R1 and concurrent R11 devices)

◆ RDF2 (R2 and concurrent R22 devices)

◆ RDF21 (cascaded R21 devices)

◆ ANY (can contain as device mix of the above types)

Note: A composite group of any SRDF type can change its type because of a `symrdf` control operation. For example, an RDF1 composite group can change to an RDF2 when the device personalities are swapped. SRDF control operations (such as the `suspend`, `establish`, and `swap` operations) cannot change the type of an ANY composite group but can affect the devices in that composite group.

SRDF consistency groups

An SRDF consistency group is a composite group comprised of SRDF devices (RDF1, RDF2, or RDF21) acting in unison to preserve dependent write consistency of a database distributed across multiple SRDF systems. If a source R1 device in the consistency group cannot propagate data to its corresponding target R2 device, data propagation from all R1 devices in the consistency group is suspended, halting all data flow to the R2 targets.

Consistency is maintained by using either Multi-Session Consistency (MSC) for SRDF/A or SRDF Enginuity Consistency Assist (RDF-ECA) for SRDF/S. For detailed information about SRDF consistency groups, refer to the *EMC Solutions Enabler SRDF Family CLI User Guide.*

A group is considered an SRDF consistency group if it is a composite group meeting all of the following criteria:

◆ Created as type RDF1, RDF2, RDF21, or ANY

◆ Contains STD devices

- Set for consistency using the `-rdf_consistency` option, which registers it with the SRDF daemon, and then enabled using the `symcg enable` command

Note: Deleting a device group from a composite group enabled for SRDF consistency causes the SRDF daemon to stop monitoring this composite group.

Note: For detailed interoperability information, please refer to E-Lab Interoperability Navigator which can be reached at http://elabnavigator.EMC.com.

Creating a composite group

A composite group is created by first defining a named empty group of a specific type explained in "Composite group types" on page 126. Then, devices OR device groups are added to the composite group.

Creating an empty composite group

Composite groups are created using the `symcg` command. When you create a composite group, you assign it a name and a group type.

Use the following command syntax to create a composite group:

```
symcg [-i Interval] [-c Count][-v]
   create CgName [-type REGULAR | RDF1 | RDF2 | RDF21 | ANY ]
   [-apidb | -rdf_consistency]
```

If a group type is not specified, the default group created is REGULAR.

Specify the interval and count options (`-i` and `-c`) to wait a predetermined time (interval) between attempts (count) to acquire an exclusive lock on the host database and, for SRDF control operations, on the local and/or remote arrays.

Creating a composite group from devices in a storage group

Starting with Solutions Enabler V8.0.1, you can create add selected members of a storage group to a target composite group. If the composite group does not exist, it is created. If none of the optional device types are specified, the default is to add standard devices.

Use the following syntax to create a composite group from specified storage group devices:

```
symsg -sid <SymmID> [-i <Interval>] [-c <Count>] [-v]
 . . .
         sg2cg <SgName> <CgName> [-bcv | -vdev | -tgt]
               [-R1 | -R2 | -R21 | -noRDF] [-apidb | -rdf_consistency]
```

If the storage group is a cascaded storage group, `sg2cg` creates a composite group for the parent storage group with devices from each of the child storage groups.

Creating composite groups with SRDF consistency

A composite group of type RDF1, RDF2, or RDF21 is registered with the SRDF daemon when the `-rdf_consistency` option parameter is supplied with `symcg create`.

Note: A composite group created with SRDF consistency must be enabled using the `symcg enable` command before the SRDF daemon can begin monitoring and managing it.

Note: The `symcg enable` command will be blocked if the type of the enable being performed is MSC or SRDF-ECA and the scope of the enable contains multiple SRDF Groups.

When creating a composite group with SRDF consistency, the composite group name is compared against all existing composite group names for uniqueness. This comparison is not case sensitive, ensuring there are no composite group naming collisions in the Symmetrix File System (SFS).

For example, to create a composite group named `mycg1` of type RDF1 with consistency enabled, enter:

```
symcg create mycg1 -rdf_consistency -type rdf1
```

In the options file, you must set the following option to `ENABLE` to create composite groups with SRDF consistency:

```
SYMAPI_USE_RDFD=ENABLE
```

Note: If devices set for consistency protection are in an existing composite group, you cannot add them to another composite group enabled for consistency protection. However, you can add these devices to a composite group not enabled for consistency protection.

Setting controls on Celerra devices

Use the following `-celerra` option to set the `rw_enble`, `write_disable`, `ready`, and `non_ready` controls on Celerra FBA devices in a composite group:

```
symcg -cg CgName [-i Interval] [-c Count]
   [-noprompt] [-v] [-force]
   [-bcv | -vdev | -tgt] [-star][-sid SymmID]
   [-celerra]
```

Exporting a composite group to file

Use the following syntax to export a composite group to file:

```
symcg [-i Interval] [-c Count][-v]
        export <CgName> [-file <FileName>] [-rdf]
          [-grpfile <GrpDbFileName>]

        exportall [-file <FileName>] [-rdf]
          [-grpfile <GrpDbFileName>]
```

Specify the interval and count options (`-i` and `-c`) to wait a predetermined time (interval) between attempts (count) to acquire an exclusive lock on the host database and, for SRDF control operations, on the local and/or remote arrays.

Note: The `-rdf` option cannot be used on REGULAR or cascaded SRDF composite groups or when the composite group is of type ANY. Specifying the `-rdf` option produces an error if any Hop-2 devices are detected in the composite group.

If `-rdf` is specified, the remote partners of the STD devices and BCV devices are added to the file instead of those devices that exist in the current local composite group. The RBCV devices will become local BCVs. Non-SRDF BCVs, VDEVs, BRBCVs, and RRBCVs will be ignored in this case. The resulting file will have as many device description lines as the composite group has members.

Deleting a composite group

Use the following syntax to delete an existing composite group:

```
symcg [-i Interval] [-c Count][-v]
   delete CgName [-force] [-symforce]
```

If the composite group has members, the command fails unless `-force` is used. If `-force` is specified, the device members of the group are removed, and the group is deleted.

If the composite group is enabled for SRDF consistency, you must use `-force` to delete it.

Importing a composite group

Use the following syntax to import a composite group from a previously generated file:

```
symcg [-i Interval] [-c Count][-v]
   import CgName [-f FileName]
   [-apidb | -rdf_consistency] [-rename]

   importall [-f FileName] [-rdf_consistency]
```

When the`-rdf_consistency` parameter is specified, the composite group is registered with the SRDF daemon.

If `-rename` is specified, the devices are given the next available default device name when added to the group. If `-rename` is not used, the devices are added with the name specified in the import file. This may result in a failure if a device already has that name in the composite group.

Adding devices to a composite group

Once a composite group is created, you can add standard devices or device groups to it.

Standard devices

Use the following syntax to add a standard device to a composite group:

```
symcg [-i Interval] [-c Count][-v]
   add pd PdevName [LdevName] -cg CgName
```

or

```
symcg -cg CgName -sid SymmID
      [-i Interval] [-c Count] [-v]
      [-rdf | -hop2]
      [-rdfg GrpNum [-remote_rdfg RemoteGrpNum]]

   add dev SymDevName [LdevName] [-vdev | -tgt]
```

When adding devices by specifying a device name, the array's ID (`-sid`) is required and additional options provide greater controls. Specify the interval and count options (`-i` and `-c`) to wait a predetermined time (interval) between attempts (count) to acquire an exclusive lock on the host database on the local and/or remote arrays.

Composite groups allow remote BCV devices (RBCVs) on both links to be associated with the SRDF group. These RBCVs may be R1 or R2 type BCVs and may also have a remote BCV (RRBCV) associated behind each.

When the `-vdev` option is specified, the devices are added to the virtual device list. For this action to succeed, the device must already be configured as a VDEV. In addition, to add a VDEV to the remote virtual device list, specify the `-rdf` option with the `-vdev` options. If the device is two hops away, specify the `-hop2` option instead.

When the `-tgt` option is specified, the devices are added to the target device list. In addition, to add a target device to the remote target device list, specify the `-rdf` option with the `-tgt` options. If the device is two hops away, specify the `-hop2` option instead.

All devices

Use the following syntax to add devices to a specified composite group. By specifying `-noRDF`, you can prohibit any SRDF devices from becoming members of the composite group.

```
symcg -cg CgName [-i Interval] [-c Count] [-v]
        [-sid SymmID]
        [-SA # | ALL] [-P #] [-N #]
        [-cap # [-captype mb | cyl]]
        [-rdf | -hop2] [-vdev | -tgt]
        [-rdfg GrpNum [-remote_rdfg RemoteGrpNum]]
        [-sel_rdfg SelRdfGrpNum]
        [-devs SymDevStart:SymDevEnd | SymDevName
            [,SymDevStart:SymDevEnd | SymDevName...]]

   addall [pd | devs [-rdf | -hop2]]
        [-R1 | -R2 | -R21 | -noRDF] [-v]
```

Adding and removing device groups

A device group does not have to contain devices before adding it to a composite group.

Use the following syntax to add one or more device groups to a composite group.

```
symcg -cg CgName [-i Interval] [-c Count][-v]
   add dg DgName[DgName1,DgName2,…,DgName10]
```

Note: The command fails if the device group contains one or more storage groups.

Use the following syntax to remove one or more device groups from a composite groups.

```
symcg -cg CgName -sid SymmID
        [-i Interval] [-c Count] [-v]
        [-rdf | -hop2]
        [-rdfg GrpNum [-remote_rdfg RemoteGrpNum]]

   remove dg DgName[DgName1,DgName2,…,DgName10]
```

The `-force` option is required if removing a device group that changes the integrity of the scope of the composite group enabled. For example, if CG1 is enabled at the composite group level, no device group can be removed from it without `-force`. If CG1 is enabled only at the SRDF group name level, any device group that is not part of that group name may be removed without using `-force`.

Moving and copying devices

Devices from an existing composite group can be moved or copied into another existing composite group of compatible type. When the `move ld` or `moveall` action is used, the devices are removed from the source composite group and added to the destination composite group. When the `copy ld` or `copyall` action is used, copies of the devices are added to the destination composite group, and the source composite group remains unchanged. You can assign a new name to a device being copied or moved. Note that by specifying the `-sel_rdfg` option, only SRDF devices belonging to that group number are moved or copied. By using this option with `-R1`, `-R2`, `-R21`, or `-noRDF`, you can further limit the number of SRDF devices added to a composite group.

Use the following syntax to move or copy devices from one composite group to another.

```
symcg -cg CgName [-i Interval] [-c Count][-v]

   move ld LdevName DestCgName [-force] [-rename]
   copy ld LdevName DestCgName [-force] [-rename]

symcg -cg CgName [-i Interval] [-c Count] [-v]
          [-sid SymmID]
          [-SA # | ALL] [-P #] [-N #]
          [-cap # [-captype mb | cyl]]
          [-vdev | -tgt [-hop2] | -rvdev | -rtgt]
          [-rdfg GrpNum [-remote_rdfg RemoteGrpNum]]
          [-sel_rdfg SelRdfGrpNum]
          [-devs SymDevStart:SymDevEnd | SymDevName
              [,SymDevStart:SymDevEnd | SymDevName...]]

       moveall DestCgName [-force] [-symforce] [-rename]
           [-R1 | -R2 | -R21 | -noRDF]

       copyall DestCgName [-force] [-symforce]
           [-R1 | -R2 | -R21 | -noRDF]
```

Note: The `symcg copy` and `symcg copyall` commands do not allow you to copy any devices from a composite group containing device groups. In this case, use the `symdg copy` or `symdg copyall` command to copy devices from a device group.

Copying devices from a device group to a composite group

Devices from an existing device group can be copied (or added) to a composite group. If the composite group does not exist, it is created.

Use the following syntax to add devices from an existing device group to an existing composite group:

```
dg2cg DgName CgName [-rename] [-force]
   [-bcv [-hop2] | -nobcv | -rbcv | -rrbcv | -brbcv |
   -vdev [-hop2] | -rvdev | -tgt [-hop2] | -rtgt]
   [-apidb | -rdf_consistency]
```

The following additional options are available:

- `-bcv` adds only the BCVs to the composite group. If the `-hop2` option is specified, it indicates that the specified device is two hops away.
- `-vdev` adds only the virtual devices to the composite group. If the `-hop2` option is specified, it indicates that the specified device is two hops away.
- `-rbcv` adds only the RBCVs to the composite group.
- `-rrbcv` adds only the RRBCVs to the composite group.
- `-brbcv` adds only the BRBCVs to the composite group.
- `-nobcv` adds only the STDs to the composite group.
- `-rvdev` adds only the remote virtual devices to the composite group.
- `-tgt` adds only the TGTs to the composite group. If the `-hop2` option is specified, it indicates that the specified device is two hops away.
- `-rtgt` adds only the RTGTs to the composite group.

By default, all device lists from the device group are added to the composite group.

Removing standard devices

Devices may be removed from an existing composite group. You can either remove devices individually, remove all devices, or remove all devices meeting the specified criteria.

Use the following syntax to remove a standard device from a composite group:

```
symcg -cg CgName -sid SymmID
   [-i Interval] [-c Count] [-v]
   [-rdf | -hop2]
   [-rdfg GrpNum [-remote_rdfg RemoteGrpNum]]
   remove dev SymDevName [-force] [-symforce]
      [-vdev | -tgt]
```

or

```
symcg -cg CgName [-i Interval] [-c Count] [-v]
remove ld LdevName [-force] [-symforce]
```

or

```
symcg -cg CgName [-i Interval] [-c Count] [-v]
remove [pd] PdevName [-force] [-symforce]
```

When the SRDF group option, `-sel_rdfg GrpNum`, is specified, the devices in that group are removed from the composite group.

Use the following syntax to remove all standard devices from a composite group:

```
symcg -cg CgName [-i Interval] [-c Count] [-v]
   [-sid SymmID]
   [-SA # | ALL] [-P #] [-N #]
   [-cap # [-captype mb | cyl]]
   [-rdf | -hop2] [-vdev | -tgt]
   [-rdfg GrpNum [-remote_rdfg RemoteGrpNum]]
   [-sel_rdfg SelRdfGrpNum]
   [-devs SymDevStart:SymDevEnd | SymDevName
   [,SymDevStart:SymDevEnd | SymDevName...]]

   rmall [-rdf | -hop2] [-force] [-symforce]
      [-R1 | -R2 | -R21 | -noRDF]
```

Removing an consistency-enabled device does not disable the device. You must use `symcg disable` to disable the composite group. If SRDF consistency is enabled and cannot be disabled, use `-symforce`.

Renaming a composite group

To rename a composite group, enter:

```
symcg [-i Interval] [-c Count]
   rename OldCGName NewCGName
```

Note: You cannot rename an SRDF consistency composite group that is enabled.

Setting SRDF group attributes

Use the following `set` syntax to associate a logical name or an SRDF/Star recovery SRDF group number with an SRDF group.

```
symcg -cg CgName [-i Interval] [-c Count] [-v]
   set -name [Name] | -recovery_rdfg GrpNum
   [-rdfg SymmID:GrpNum[,GrpNum,...]|all[,...] |
      name:RdfGroupName[,RdfGroupName]]
```

This command provides a mechanism to group multiple RA groups within a specified composite group to perform operations on all associated devices by specifying the SRDF group name. If the `-rdfg` argument is not specified, then the action is applied to all SRDF groups.

Composite group creation and output example

The `symcg` command can be used to obtain information about the composite groups visible to your host system. Use the following syntax to list the host-visible composite groups:

```
symcg list
```

The `-inactive` option is available for use in GNS-enabled environments to return the list of composite groups from the inactive group definition list. For information on active and inactive group lists in GNS, refer to "Active vs. inactive group lists" on page 333.

The following example creates a nonconsistent RDF1 composite group:

```
symcg create MyNewCG -type rdf1
```

RDF1 devices are added to our `MyNewCG` example:

```
symcg -cg MyNewCG addall -devs CEE:CFD -sid 79
symcg -cg MyNewCG addall -devs 2:7 -sid 32
```

Then, RBCV devices are associated with the `MyNewCG` composite group:

```
symbcv -cg MyNewCG associateall -devs E92:EA1 -rdf -rdfg 64 -sid 79
symbcv -cg MyNewCG associateall -devs 48:4D -rdf -rdfg 1 -sid 32
```

In addition, RRBCV devices are associated with the `MyNewCG` composite group:

```
symbcv -cg MyNewCG associateall -devs 328:337 -rrdf -rdfg 64 -sid 79
symbcv -cg MyNewCG associateall -devs 30:35 -rrdf -rdfg 1 -sid 32\
```

As a result, the `symcg list` command returns the following output:

```
symcg list

                    C O M P O S I T E    G R O U P S

                    Number of            Number of
  Name   Type   Valid  Symms RAGs  DGs  Devs   BCVs   VDEVs    TGTs

MyNewCG RDF21   Yes       1    2    2     3      0       0       0
```

Note that this command returns the number of device groups contained in the MyNewCG composite group.

Use the `show` action to return additional details about the `MyNewCG` composite group.

Displaying composite group information

Use the following syntax to list the available composite groups:

```
symcg [-i Interval] [-c Count][-v]
   list [-offline] [-v]
```

```
symcg list

                    C O M P O S I T E    G R O U P S

                    Number of            Number of
  Name   Type   Valid  Symms RAGs  DGs  Devs   BCVs   VDEVs    TGTs

  cgr21 RDF21   Yes       1    2    2     3      0       0       0
  cgreg REGULAR Yes       0    0    0     0      0       0       0
```

Note: Composite groups with names longer than 21 characters display with their first 21 characters followed by an asterisk (*). "Setting group names to display full names" on page 31 explains how to set the environmental variable to display full group names.

Use the `show` action to return additional details about a composite group. For example, to show the details about composite group `cgr21`, enter:

```
symcg show cgr21

Composite Group Name:  cgr21

  Composite Group Type                              : RDF21
  Valid                                             : Yes
  CG in PowerPath                                   : No
  CG in GNS                                         : No
  RDF Consistency Protection Allowed                : No
  RDF Consistency Mode                              : NONE
  Concurrent RDF                                    : YES
  Cascaded RDF                                      : No

  Number of RDF (RA) Groups                         :    2
  Number of STD Devices                             :    3
  Number of CRDF STD Devices                        :    3
  Number of BCV's (Locally-associated)              :    0
  Number of VDEV's (Locally-associated)             :    0
  Number of TGT's Locally-associated                :    0
  Number of CRDF TGT Devices                        :    0
  Number of RVDEV's (Remotely-associated VDEV)      :    0
  Number of RBCV's (Remotely-associated STD-RDF)    :    0
```

```
  Number of BRBCV's (Remotely-associated BCV-RDF)    :    0
  Number of RRBCV's (Remotely-associated RBCV)       :    0
  Number of RTGT's (Remotely-associated)             :    0
  Number of Hop2 BCV's (Remotely-assoc'ed Hop2 BCV)  :    0
  Number of Hop2 VDEV's (Remotely-assoc'ed Hop2 VDEV):    0
  Number of Hop2 TGT's (Remotely-assoc'ed Hop2 TGT)  :    0
  Number of Device Groups                            :    2
  Device Group Names                                 : dg1
                                                     : dg2

  Number of Symmetrix Units (1):

    1) Symmetrix ID                                   : 000194900341
       Microcode Version                              : 5874
       Number of STD Devices                          :    3
       Number of CRDF STD Devices                     :    3
       Number of BCV's (Locally-associated)           :    0
       Number of VDEV's (Locally-associated)          :    0
       Number of TGT's Locally-associated             :    0
       Number of CRDF TGT Devices                     :    0
       Number of RVDEV's (Remotely-associated VDEV)   :    0
       Number of RBCV's (Remotely-associated STD_RDF) :    0
       Number of BRBCV's (Remotely-associated BCV-RDF):    0
       Number of RTGT's (Remotely-associated)         :    0
       Number of RRBCV's (Remotely-associated RBCV)   :    0
       Number of Hop2BCV's (Remotely-assoc'ed Hop2BCV):    0
       Number of Hop2VDEVs(Remotely-assoc'ed Hop2VDEV):    0
       Number of Hop2TGT's (Remotely-assoc'ed Hop2TGT):    0

       Number of RDF (RA) Groups (2):


         1) RDF (RA) Group Number         :  11            (0A)
            Remote Symmetrix ID           : 00019490237
            Microcode Version             : 5874
            Recovery RA Group             : N/A            (N/A)
            RA Group Name                 : N/A

            STD Devices (3):

              -------------------------------------------------------
                                      Sym  Device        Flags  Cap
              LdevName      PdevName   Dev  Config     Sts CSRT  (MB)
              -------------------------------------------------------
              DEV341_30     /dev/sdl   0030 RDF21+R-5  WD  X--2  2063
              DEV341_31     /dev/sdm   0031 RDF21+R-5  WD  X--2  2063
DEV341_E6     N/A        00E6 RDF21+R-5  WD  X--2  2063

         2) RDF (RA) Group Number         :  18            (11)
            Remote Symmetrix ID           : 00019490016
            Microcode Version             : 5874
            Recovery RA Group             : N/A            (N/A)
            RA Group Name                 : N/A

            STD Devices (3):
            CRDF STD Devices (3):

              -------------------------------------------------------
                                      Sym  Device        Flags  Cap
              LdevName      PdevName   Dev  Config     Sts CSRT  (MB)
              -------------------------------------------------------
              DEV341_30     /dev/sdl   0030 RDF21+R-5  WD  XAM1  2063
              DEV341_31     /dev/sdm   0031 RDF21+R-5  WD  XAM1  2063
              DEV341_E6     N/A        00E6 RDF21+R-5  WD  XAM1  2063
```

Note: The displayed logical device names are truncated to an existing standard length of eleven characters.

Showing composite groups associated with a device group

The following output shows that the `dgincg` device group is a member of two composite groups, `cg110` and `cgregular`. Also, `dgincg` is a group type of `ANY`.

To show the composite groups containing the `dgincg` device group, enter:

```
symdg show dgincg

Group Name:  dgincg

    Group Type                                    : ANY
    Device Group in GNS                           : No
    Valid                                         : Yes
    Symmetrix ID                                  : 000194900341
    Group Creation Time                           : Tue Dec 1 8:6:9 2009
    Vendor ID                                     : EMC Corp
    Application ID                                : SYMCLI

    Number of STD Devices in Group                : 2154
    Number of Associated GK's                     :    0
    Number of Locally-associated BCV's            :  128
    Number of Locally-associated VDEV's           :    0
    Number of Locally-associated TGT's            :    0
    Number of Remotely-associated VDEV's(STD RDF):    0
    Number of Remotely-associated BCV's (STD RDF):    0
    Number of Remotely-associated TGT's(TGT RDF) :    0
    Number of Remotely-associated BCV's (BCV RDF):    0
    Number of Remotely-assoc'd RBCV's (RBCV RDF) :    0
    Number of Remotely-assoc'd BCV's (Hop-2 BCV) :    0
    Number of Remotely-assoc'd VDEV's(Hop-2 VDEV):    0
    Number of Remotely-assoc'd TGT's (Hop-2 TGT) :    0
    Number of Composite Groups                    :    2
    Composite Group Names                         : cg110
                                                  : cgregular
```

Performing miscellaneous operations on composite group devices

Use the following command syntax to perform the following operations on devices within a composite group:

- Change the state (ready, not ready) of a device.
- Set or reset the hold bit on a device.
- Compress or uncompress a thin device
- Bind, unbind, rebind, start or stop allocations, free, and reclaim thin devices

Note: For Enginuity versions 5876 or lower, free and reclaim actions are blocked for any device which is the source or target of a Clone, Emulation, or Snap session.

- Set or unset the persistence of a device
- Set Optimized Read Miss mode (ORM)
- Set or unset Geometry Compatible Mode (GCM)

For arrays running HYPERMAX OS 5977, the Optimized Read Miss feature is not supported.

```
symcg -cg CgName [-i Interval] [-c Count] [-v]
          [-noprompt] [-force]
          [-bcv [-hop2] | -rbcv | -brbcv | -rrbcv |
           -vdev [-hop2] | -rvdev | -tgt [-hop2] | -rtgt ]
          [-rp] [-star] [-sid SymmID |
          -rdfg SymmID:GrpNum[,GrpNum,...]|all [,...] |
            name:RdfGroupName [,RdfGroupName]] [-celerra]

        not_ready [LdevName [LdevName ...]]

        ready [LdevName [LdevName ...]]

        hold [LdevName [LdevName ...]]

        compress [LdevName [LdevName ...]] [-stop]

        uncompress [LdevName [LdevName ...]] [-stop]

        bind [LdevName [LdevName...]] -pool <PoolName>

        rebind [LdevName [LdevName...]] -pool <PoolName>

        unbind [LdevName [LdevName...]]

        allocate [LdevName [LdevName...]] [-persistent | -stop]

        free [-all] [LdevName [LdevName...]] [-stop]

        reclaim [LdevName [LdevName...]] [-persistent | -stop]

        set -persistent [LdevName [LdevName ...]]

        unset -persistent [LdevName [LdevName ...]]

        set -orm [on |off | system] [LdevName [LdevName...]]

symcg -cg <CgName> [-i <Interval>] [-c <Count>] [-v]
          [-noprompt] [-symforce]
          [-bcv [-hop2] | -rbcv | -brbcv | -rrbcv |
           -vdev [-hop2] | -rvdev | -tgt [-hop2] | -rtgt ]
          [-rp] [-star] [-sid <SymmID> |
          -rdfg <<SymmID>:<<GrpNum>[,<GrpNum>,...]|all>[,...] |
            name:<RdfGroupName>[,<RdfGroupName>]>] [-celerra]

        unhold [<LdevName> [<LdevName> ...]]

        set -gcm [<LdevName> [<LdevName> ...]]

        unset -gcm [<LdevName> [<LdevName> ...]]
```

Note: "Showing details about thin devices" on page 465 provides information on compressing and uncompressing thin devices.

For arrays running Enginuity 5876 Q22013SR the `symcg` command supports the following control actions when issued against thin devices: binding, unbinding and rebinding, the ability to start and stop allocations, and freeing and reclaiming of space on a thin device. The operations can be directed at all devices in a composite group. By default the actions only apply to the standard devices in the group. If a control operation needs to be performed on devices in a specific list, the appropriate qualifier needs to be used (such as, `-bcv`, `-rbcv`, etc.).

For more information on actions for thin devices (bind, unbind, rebind, allocate, etc.) refer to "Performing actions on thin devices (Enginuity 5773 through 5876)" on page 284.

For arrays running Enginuity 5876 Q22013SR, the `symcg` command allows you to set the Optimized Read Miss mode to Off or System Managed for devices in a composite group. By default, the action is only applied to standard devices in the group. If the action needs to be performed on devices in a specific list, the appropriate qualifier needs to be used (such as, `-bcv`, `-tgt`, etc.). "Optimized Read Miss mode (Enginuity 5876 only)" on page 439 provides additional information on this feature.

By using the `set -host_id` option, you can modify the host ID of devices in a composite group. Arrays running HYPERMAX OS 5977 are assigned a new device host ID format, a Federated ID that is globally unique and transportable between storage systems. When creating device groups, you can choose a host ID type that is either the new Federated ID format (`native`) or an ID compatible with the previous ID format (`compatible`) that is a non-portable ID value only unique within the array. The following restrictions apply:

- When `compatible` is specified, if the device number of any of the devices in the storage group is greater than 0xFFFF, the operation will fail.
- The operation will fail if any of the devices are online.
- The operation will fail if any of the devices are visible to a host through either mapping or masking.
- The operation will fail if `native` is specified and the emulation type of the device is AS/400_D910_099 on an array running HYPERMAX OS 5977.

Using the GNS repository

Group name services (GNS) provides a common repository to store and maintain SYMAPI device group and composite group definitions across storage arrays that are visible to all locally attached hosts. Any host with GNS enabled can view and use the device and composite groups that are contained in the repository regardless of what host actually ran the commands to create them. This provides redundancy in case of a failure that causes the host that normally runs Solutions Enabler to lose access to the VMAX array. It also creates standards for device groups and device names that can be shared across all hosts in the environment.

Starting with Solutions Enabler V8.0.1, groups are no longer stored in a SYMAPI database file but are stored either in a local database file on the host or in a global database on VMAX or DMX arrays. Both the local and global group databases are managed by the GNS daemon. The setting of SYMAPI_USE_GNS defines the group management operation mode that governs where the groups are stored:

- If SYMAPI_USE_GNS is set to DISABLE, all group information by default will be stored to a local file located in the secure directory:

  ```
  /var/symapi/gns/storgnsd.db
  ```

  where the GNS daemon has write privileges. This file is referred to as the "local group database" or simply "local file" for groups in this document.
- If SYMAPI_USE_GNS is set to ENABLE, all group information will be stored in a global database in VMAX or DMX arrays.

By default, SYMAPI_USE_GNS is set to DISABLE so the group information is stored in a local database file on the host that created the group.

Enabling GNS allows group definitions to be stored on the array in a shared GNS repository. This shared GNS repository is visible to any GNS-enabled locally-attached host, enabling these hosts to perform control operations, regardless of which host initially defined the group. In addition, if one host goes down, you can still perform SYMCLI control operation from another local host in your array environment.

Solutions Enabler SYMAPI and SYMCLI do not directly access the shared repository. Instead, requests are forwarded to the GNS daemon, which processes all GNS operations. This daemon is the only entity that directly accesses the GNS shared repository and is responsible for ensuring that each host has access to the most current GNS definitions.

From each host, a GNS daemon listens for GNS requests from local clients (same host) and carries them out on the locally attached array. In addition, the GNS daemon monitors the GNS repositories on all locally-attached arrays, at a user-configured polling interval, for changes made to the shared GNS repository by other daemons (on other hosts). When a change is identified, the GNS daemon updates the host to ensure that all GNS-enabled hosts refer to the same group definitions. A set of options are available for controlling the GNS daemon. For information on configuring the GNS daemon, refer to "GNS daemon options file" on page 147.

Note: If User Authorization is enabled, configuration and management of GNS requires a minimum role of StorageAdmin.

Automatic upgrade

When upgrading to Solutions Enabler V8.0.1 from an earlier version of Solutions Enabler, all active groups are automatically migrated from the existing SYMAPI database (either default or alternate) to a GNS-managed database. This migration happens once for a SYMAPI database.

Shared group definitions with GNS

A *device group* is a user-defined object comprised of devices that belong to a single array and RA groups on that device. In a GNS-enabled environment, device group definitions are stored in the GNS repository of the array on which the devices within the device group reside and are visible to all hosts locally attached to that array.

A *composite group* is also a user-defined object comprised of devices. However, the device members of a composite group can belong to multiple arrays and RA groups. In a GNS-enabled environment, composite group definitions are distributed across all arrays that contain device members of the composite group by the GNS daemon. A host must be locally attached to all arrays containing devices in the composite group to manage or control that composite group. If a host is only attached to a subset of the arrays that a composite group spans, that group will be visible to the host, but in an invalid and unusable state. This is not a recommended configuration.

The GNS state that is visible from a given host is determined by the set of arrays to which the host is connected. As seen in Figure 3 on page 140, Host-1 is GNS-enabled and locally attached to array A, array B, and array C. Host-1 can access and modify the group

definitions on all three arrays. The group definitions that are visible to Host-1 include device groups DG1, DG2, DG3, DG4, DG5, and DG6 and composite group CG1's definition, which contains devices on arrays B and C.

Host-2 is GNS-enabled and locally attached to array B and array C. The group definitions that are visible to Host-2 include device groups DG3, DG4, DG5, and DG6 and composite group CG1. Since Host-2 is locally attached to only array B and C, it cannot see any device groups or composite groups on array A (that is, DG1 and DG2).

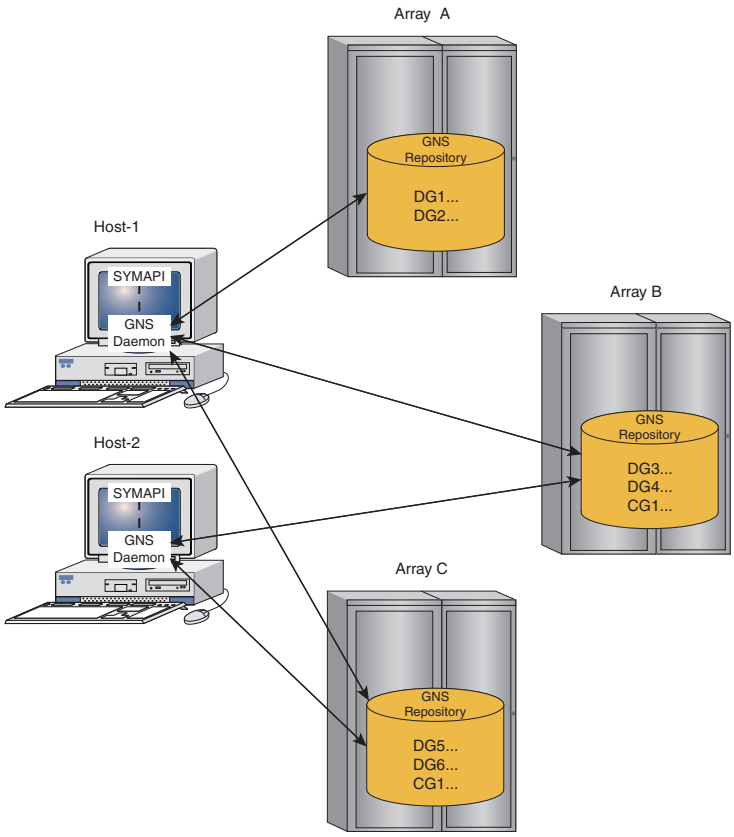


Figure 3 Host-visible GNS state

How GNS handles updates to group definitions

When GNS is enabled, device group and composite group definitions are stored in the common GNS repository.

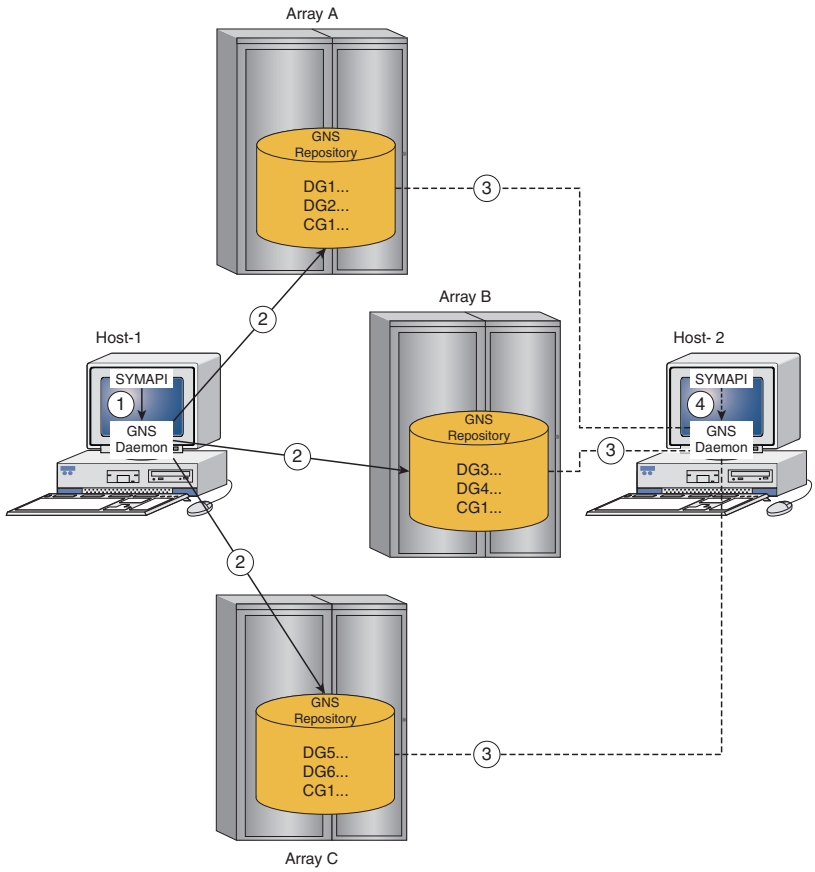


Figure 4 GNS across multiple hosts

For example, in Figure 4, Host-1 makes a change to composite group CG1.

1. The Host-1 GNS daemon gets this request to process.
2. The Host-1 GNS daemon updates the GNS-shared repository on all relevant arrays, in this case arrays A, B, and C.
3. The GNS daemon running on Host-2 detects the change while polling its locally-attached GNS repositories (attached arrays A, B, and C) and updates Host-2 with the updates made by Host-1.
4. Host-2's client application will detect the change on its next group call.

GNS and consistency groups

An SRDF consistency group is a composite group comprised of SRDF devices (RDF1, RDF2, or RDF21) acting in unison to preserve dependent write consistency of a database distributed across multiple SRDF systems. It maintains this consistency by using either Multisession Consistency (MSC) for SRDF/A or SRDF Enginuity Consistency Assist (SRDF-ECA) for SRDF/S. Both use the SRDF daemon to maintain SRDF consistency.

In a GNS-enabled environment, certain changes made to any composite group set for consistency are automatically propagated to the SRDF daemon on all relevant hosts, such as changes to the group type, SRDF group name, recovery RA group number, device membership, device group membership, and device LdevName. When updates are made to the GNS repository, the GNS daemon updates the SRDF daemon.

For more information on setting SRDF consistency to composite groups, refer to "Creating a composite group from devices in a storage group" on page 127.

GNS device groups and SRDF

In an SRDF scenario, a local GNS-backed device group (either RDF1 or RDF2) can be automatically mirrored on the remote array through SRDF links — for activation and use during a disaster fail-over situation. By default, the GNS device group definitions are stored on the local (directly attached) array — where the local devices are located.

Optionally, any changes made to the local group definition (for example, adding a standard or BCV device) can be set to automatically maintain a mirrored group definition on the remote array. This remotely mirrored group is for disaster recovery situations, where entire applications (including group aware ones), fail-over and are restarted on the remote side (where the remote devices are).

As changes are made to the local group definition, GNS automatically propagates corresponding changes to the remote array, so that the two are kept synchronized. For more information on configuring this option, refer to "Mirroring remote device group definitions" on page 148.

GNS behavior in client/server mode

In client/server mode, the SYMAPI server's SYMAPI options file setting SYMAPI_USE_GNS must be enabled to use GNS.

Active vs. inactive group lists

When GNS is enabled, group definitions are stored in the global GNS repository in addition to individual group database files. To facilitate the importing of groups into GNS, a limited access to the group definitions held within a group database is provided while GNS is enabled.

You can list the groups within the individual group database files using the following commands:

```
symdg list -inactive
symcg list -inactive
```

You can import a specific group contained in the group database into GNS via the following commands:

```
symdg activate DgName
symcg activate CgName
```

You can import all groups in the group database into GNS using the following commands:

```
symdg activateall
symcg activateall
```

Note: Importing groups into GNS may fail if the group name is already in use within GNS.

Configuring GNS

To take advantage of the GNS features, you must first configure your environment to support GNS.

Setting up the GNS option

GNS can be either enabled or disabled on each host through the SYMAPI options file setting `SYMAPI_USE_GNS`. This option is enabled as follows:

```
SYMAPI_USE_GNS=ENABLE
```

The default setting is `DISABLE`. When `SYMAPI_USE_GNS` is set to `DISABLE`, the GNS daemon stores groups in a local database file can be designated by the `SYMCLI_DB_FILE` environment variable. If the file is not specified, the default is used.

If `SYMAPI_USE_GNS` is set to `ENABLE`, all group information is stored in a global database on each array with which specific groups are associated. Note that the GNS daemon may store groups either in the global group database or in a local group database depending on whether `SYMCLI_DB_FILE` is pointing to a private database. If `SYMCLI_DB_FILE` is not set, then the global group database will be used to store groups. Otherwise, the local group database designated by `SYMCLI_DB_FILE` will be used to store groups. It is assumed that applications running with a private configuration database file intend to keep their modifications private, and therefore, store the groups in a private group database file.

If `SYMAPI_USE_GNS` is not set and if `SYMCLI_DB_FILE` is not pointing to a private database, the GNS daemon uses `/var/symapi/gns/storgnsd.db` to store groups. When you specify an alternate database, the name of the alternate group database file is derived from the value of alternate database name and the group database file is placed under the `/var/symapi/gns` directory. For example, if the alternate SYMAPI database name is `/usr/application/my_symapi_db.bin`, the alternate group database file is `/var/symapi/gns/my_symapi_db.bin_<integer>.db`. The integer is a number derived from the original path name.

Configuring GNS in a multihost environment

When configuring GNS in a multihost environment, it is recommended that you first review the groups currently stored on all hosts that share access to a set of arrays. Resolve any potential conflicts with other hosts. Also, if using composite groups, ensure that all hosts on which you intend to enable GNS are local to the same set of arrays to avoid composite groups appearing invalid to hosts that are not attached to all referenced arrays.

When ready, enable GNS on a single array being sure to configure any options or user authentication that you desire. For details on managing GNS use through the options file, refer to the *EMC Solutions Enabler SYMCLI Command Reference Guide*. In addition to enabling GNS on the host, you can configure GNS User Authentication (set up a specific set of users with access to the GNS daemon) and GNS Daemon Control (start and stop the daemon, and configure GNS daemon options).

Once GNS is enabled on a host, activate those groups that you wish to expose using GNS. Enable GNS on a second host. Validate that the new host can see the previously activated groups and resolve any conflicts with other hosts before activating additional groups.

Note: With Solutions Enabler V7.4 and higher, device and composite groups of type REGULAR are allowed to contain remote SRDF devices. However, on systems running GNS, any peer hosts running older versions of Solutions Enabler sees these groups as invalid.

GNS user authentication

GNS is intended to share group definitions across multiple users and hosts in a environment via the host-installed GNS daemons. As such, access to GNS functionality is controlled by limiting permission to the GNS daemon. This access is controlled through the common daemon authorization file, `daemon_users`. This file is located in the following directories:

UNIX	`/var/symapi/config/daemon_users`
Windows	`c:\Program Files\EMC\SYMAPI\config\daemon_users`

Note that non-root Solutions Enablers users need to modify `c:\Program Files\EMC\SYMAPI\config\daemon_users` or `/var/symapi/config/daemon_users` to authorize the use of the GNS daemon.

Note: It is important to protect this file so that only privileged administrators can modify it.

Users meeting any of the following criteria will be permitted to control and use the GNS daemon:

- User with privileges; UNIX users with root access and Windows users that are a members of the Administrators group
- Users listed in the `daemon_users` file located on each host from which they require access

For example, the following lines in the `daemon_users` file would permit users named `Smith` and `Jones` to use the GNS daemon:

```
smith    storgnsd
jones    storgnsd
```

Executing the `Stordaemon setuser` command for specified users grant those users permission to use all daemons. For example:

```
Stordaemon setuser  all -user smith
```

GNS daemon control

There are three ways that the GNS daemon can be started.

- The daemon will be started automatically by the Solutions Enabler libraries the first time they attempt to connect with it, which can cause a slight delay in performance on that initial connection while the daemon starts and builds its cache.

Note: Prior to starting `storgnsd`, ensure that your default group database is up-to-date, since `storgnsd` uses the information stored in it to establish contact with your arrays.

- You can start the daemon manually using the `stordaemon` command line utility as follows:

  `stordaemon start storgnsd`

- You can set the daemon to start automatically every time the local host is booted using the following command line:

  stordaemon install storgnsd -autostart

Prestarting the daemon, either manually or using the automatic option, is useful because the daemon may take a while to initially construct its cache — depending on the number of groups and arrays it has to load.

If the daemon is stopped for some reason, it can optionally be restarted automatically by an internal Solutions Enabler watchdog mechanism. A combination of the watchdog mechanism and the auto-start option described above can be used to ensure that the daemon is always running, which is important with SRDF consistency composite groups to ensure that the MSC SRDF daemon is updated whenever group changes are made.

Enabling the watchdog

This watchdog mechanism is enabled by default. The watchdog daemon will restart the GNS daemon if it crashes or is killed. This behavior can be disabled through the `storgns:autorestart` entry in the `daemon_options` file (for more information, refer to "GNS daemon options file" on page 147). On UNIX, this restart functionality is provided by a dedicated watchdog daemon (`storwatchd`). This is started automatically as needed. On Windows, this restart functionality is provided by the Service Control Manager.

Note: A maximum of three restarts within a 15-minute period will be attempted. Following a third restart (again, within a given 15-minute period), no subsequent restart will be attempted.

Managing the GNS daemon

You can control the GNS daemon with the `stordaemon` command. The control options include:

- Stopping the daemon.
- Querying the daemon's status.
- Querying the daemon's log files.

For details, refer to the `stordaemon` man page in the *EMC Solutions Enabler SYMCLI Command Reference Guide*.

Accessing groups in offline mode

To access a non-default alternate group database file (for example, from a different host), you can use one of two methods to specify the file:

1. Specify a `SYMCLI_GROUP_DB` environmental variable to indicate the path of the group database file.
2. Specify a `-grpfile` command line option for `symcg` and `symdg` commands.

   The `symcg` list, show, export and exportall command syntax is as follows:

```
symcg [-i <Interval>] [-c <Count>] [-v]
. . .
        export <CgName> [-file <FileName>] [-rdf]
          [-grpfile <GroupsDbFileName>]
```

```
exportall [-file <FileName>] [-rdf]
          [-grpfile <GroupsDbFileName>]
. . .
list [-offline] [-v]
  [-apidb | -rdf_consistency]
  [-grpfile <GroupsDbFileName>]
. . .
show <CgName> [-inactive] [-offline | -lock]
  [-grpfile <GroupsDbFileName>]
. . .
```

The symdg list, show, export and exportall command syntax is as follows:

```
symdg [-i <Interval>] [-c <Count>] [-v]
. . .
        export <DgName> [-delete] [-file <FileName>]
          [[-rdf [-rdfg <GrpNum>]] | [-sid <SymmID>]]
[-grpfile <GroupsDbFileName>]

        exportall [-delete] [-file <FileName>]
          [[-rdf [-rdfg <GrpNum>]] | [-sid <SymmID>]]
[-grpfile <GroupsDbFileName>]
. . .
        list [-sid <SymmId>] [-offline] [-v]
          [-grpfile <GroupsDbFileName>]
. . .
        show <DgName> [-inactive] [-offline | -lock]
[-grpfile <GroupsDbFileName>]
. . .
```

Both methods must specify a full path name of the alternate group database file. In addition, both methods are restricted to only symcg and symdg list, show, export, and exportall operations.

If the specified file cannot be accessed, no error will be returned. Instead, all access attempts will return a “no groups found” error message.

GNS daemon and external locks

The GNS daemon uses two External Locks to maintain exclusive access to the GNS repository on each array: F0 and F1.

Use the following command to view available GNS locks:

```
symcfg -sid nnnn -lockn GNS list
```

Use the following command to manually release a GNS lock:

```
symcfg -sid nnnn -lockn GNS release
```

GNS daemon log file

The GNS daemon writes its log (trace) messages to the standard location used by all daemons. The locations are:

UNIX `/var/symapi/log/storgnsd.log0`
`storgnsd.log1`

Windows `c:\Program Files\EMC\SYMAPI\log\storgnsd.log0`
`storgnsd.log1`

These two files are written in an alternating, round-robin manner. When the active one becomes full, it is closed and the other one is truncated and made active.

The contents of the log file can also be displayed as follows:

```
stordaemon showlog storgnsd -lines 200
```

GNS daemon options file

Configuration options for the GNS daemon are contained within the daemon options file (`storgnsd`) located in the following directories:

UNIX	`/var/symapi/config/daemon_options`
Windows	`c:\Program Files\EMC\SYMAPI\config\daemon_options`

Options can be specified in this file using the following syntax:

```
storgnsd:OptName = OptValue
```

Where the *`OptName`* is the name of the option and the *`OptValue`* is the new value you wish to set.

Editing or removing parameters manually

The daemon options file contains a set of parameters that can be modified to affect GNS behavior when using SYMCLI or SYMAPI commands. The file contains editable behavior parameters set to certain optional defaults in the line entries. Commented lines beginning with a pound sign (#) are ignored.

To remove any parameter option, remove the line entry, rename the file, or comment the line by adding a pound sign (#) at the beginning of the line entry.

Editing or removing parameters via the CLI

Option file values also can be set using the `stordaemon setoption` CLI utility, which programmatically makes changes to the `daemon_options` file. The following is the command-line syntax for this option:

```
stordaemon setoption DaemonName
-name OptName=OptValue [-force]
```

Where the *`OptName`* is the name of the option and the *`OptValue`* is the new value you wish to set.

The following example shows how this utility can be used to set or change an option setting:

```
stordaemon setoption storgnsd -name autorestart=enable
```

The following example shows how this utility can be used to remove an option setting:

```
stordaemon setoption storgnsd -name autorestart=
```

Note: In the above example an empty value has been passed.

By default, only options already present in the file (such as those in use or commented out) can be set. The `-force` option can be supplied to force a change, even if it requires adding a new option line to the file.

You can also set the daemon option value using the `stordaemon setvar` utility, which modifies the value during runtime. The following is the syntax for this option:

```
Stordaemon  setvar  Daemonname  -name Var=Value
```

Where the `Var` is the name of the option and the `Value` is the new value you wish to set. For example:

```
stordaemon setvar storgnsd -name autorestart=enable
```

The command will change the option value during runtime and will not modify the daemon options file. Upon daemon restart, the daemon option will lose the value which was set using the `setvar` command.

Note: For information on the possible GNS daemon options file parameters, refer to the *EMC Solutions Enabler SYMCLI Command Reference Guide*.

Mirroring remote device group definitions

The option `gns_remote_mirror` in the GNS daemon's options file determines whether GNS should attempt to remotely mirror a device group or a composite group SRDF definition contained in the shared GNS repository for a given array.

You can enable or disable the GNS remote mirror service at runtime without restarting GNS daemon. Note that the GNS daemon does not automatically pick up the change in the daemon_options file. You need to execute the `stordaemon reload` command to make the new option setting effective.

When enabled, GNS will maintain a remote mirrored group definition with the same name as the local one creating a usable group to hosts (including GNS daemons) directly connected to the remote arrays. The remote mirrored group has the same name as the local one, and has a mirror image of its contents; in other words, the data reflects the perspective of the local array. This done to ensure that the mirrored group is a legal, usable group to hosts directly connected to the remote arrays.

Note: SRDF for 10K Series solutions are supported with Enginuity version 5875Q32001SR and 5876. GNS supports the remote mirroring of groups between local and remote 10K arrays, connected via SRDF and between a VMAX 10K array and a non- VMAX 10K array running Enginuity 5773 or higher.

Controlling the mirrored group

An SRDF group that has been created by this mirroring mechanism is flagged internally by GNS as being a mirror.

By default, these mirrored groups are read-only and cannot be modified or renamed—although GNS will continue to update them as the local groups upon which they are based are changed. This behavior can be changed by setting the SYMAPI_GNS_MIRRORED_GROUP_CONTROL option in the SYMAPI options file to `ENABLE` (the default value is `DISABLE`).

If a mirrored group is directly modified (from a host connected to the remote array on which it is defined), the connection between it and the local group, on which it was based, is broken. At that point, it is no longer a mirror and changes to its base group (the local group) are no longer propagated to it. If a mirrored group is renamed or deleted, GNS will subsequently recreate the mirrored group from its base group.

Mirroring exceptions

Any group containing one or more of the following configurations cannot be mirrored:

- Groups containing any concurrent devices
- Groups containing any cascaded devices

- Any composite group containing a device group
- Any group containing an SRDF STD device paired with a remote BCV
- Groups containing any hop-2 devices (including 2TGT, 2BCV, 2VDEV devices)
- Device and composite groups of type=ANY
- Groups containing no standard SRDF devices

Impact of remotely mirroring group definitions

The following changes are made to the remote group's definition:

- The remote group definition maintains the same name and type as the local group.
- The group subtype is reversed: an RDF1 local group becomes an RDF2 remote one; an RDF2 local group becomes an RDF1 remote one.
- Attributes that are unrelated to particular devices or storage arrays are copied to the remote copy without change. For example, modification time, vendor, and group name.
- The remote RA group number (RRAGRP) attribute is changed on the remote side to contain the corresponding local SRDF group number.
- A number of attributes come in pairs: local and remote. On the remote side, these are swapped. For example:
  - The local and remote standard (DEVS and RDEVS), the local and remote BCV device lists (BCVs and RBCVs), and remote BCVs of the remote BCV device lists (BRBCVs and RRBCVs) are swapped.
- The device logical device name list is left unchanged. The device and remote device lists are swapped as noted above, but the logical device name list is not. The same names apply to the local device in both the original and mirrored copy.

BCV and remote BCV device alias name swap

In the remote mirror name transformation, the BCV alias list becomes the RBCV alias list in the remote mirror. Alias names that have a value of BCV*nnn* are automatically changed to RBCV*nnn*.

In turn, the RBCV alias list becomes the BCV alias list in the remote mirror. In this case the alias names that have a default value of RBCV*nnn* are automatically changed to BCV*nnn*.

Similarly, the RRBCV logical list becomes the BRBCV list in the remote mirror. Logical names that have a default value of RRBCV*nnn* are automatically changed to BRBCV*nnn*.

As expected, the BRBCV logical list becomes the RRBCV list in the remote mirror. Logical names that have a default value of BRBCV*nnn* are automatically changed to RRBCV*nnn*.

These names are the default names (logical device names) supplied by SYMCLI if the user does not provide one. If the user has overridden the defaults and supplied their own names that look like the above, they will still be transformed as described.

In the unlikely scenario that the user has explicitly supplied aliases, for example, BCV*nnn* and RBCV*nnn*, it is possible that the above transformation may result in duplicate alias names in the remote mirror. For example, if the original BCV alias list contained `BCV001,RBCV001`, the remote mirror's RBCV alias list would contain `RBCV001,RBCV001`, which would cause a name conflict.

Figure 5 illustrates the mirrored group data:

```
--- Local DG Group, on Symm1 ---          --- Remote DG Group, on Symm2 ---
Type            RDF1                      Type         RDF2
Role            Have a Remote Mirror      Role         Am a Remote Mirror
Remote Symm     Sym2                      Remote Symm  Sym1
Local RDFGrp    2                         Local raGrp  5
Remote RDFGrp   5                         Remote raGrp 2
Devices         0001,0002,0003            Devices      0091,0092,0093
Remote-Devs     0091,0092,0093            Remote-Devs  0001,0002,0003
LDevNames    DEV001,DEV002,DEV003         Dev Aliases  DEV001,DEV002,DEV003

BCVs            0010,0011                 BCVs         0020,0021,0022
Remote-BCVs     0020,0021,0022            Remote-BCVs  0010,0011
BCV LdevName    BCV001,BCV002             BCV Aliases  BCV001,BCV002,BCV003
RBCV LdevName   RBCV001,RBCV002,RBCV003   RBCV Aliases RBCV001,RBCV002

Vendor          EMC                       Vendor       EMC
ModTime         12345                     ModTime      12345
GateKeepers     /dev/rdsk/c2t0d2
```

Figure 5 Mirrored remote group definition

Identifying GNS groups

After you have configured your array environment to support the GNS option, you can begin to create and modify groups. Use `symdg show DgName`, for device groups, or `symcg show CgName`, for composite groups, to determine if a group is stored in the GNS group list.

Group output

For a device group, the output of the command `symdg show DgName` identifies whether a group definition is stored in GNS. The sample output below shows that group `MyRegDeviceGroup` is stored in the GNS list since the output parameter **`Device Group in GNS`** has a value of **`Yes`**. It also shows the device group's GNS mirror state.

```
% symdg show MyRegDeviceGroup

Group Name: MyRegDeviceGroup

    Group Type                                    : REGULAR
    Device Group in GNS                           : Yes (Is Mirror)
    Valid                                         : Yes
    Symmetrix ID                                  : 000184500160
    Group Creation Time                           : Tue Aug 4 16:44:18 2006
    Vendor ID                                     : EMC Corp
    Application ID                                : SYMCLI

    Number of STD Devices in Group                :   20
    Number of Associated GKs                     :    0
    Number of Locally-associated BCVs            :   20
    Number of Locally-associated VDEVs           :    0
. . .
```

In addition, for a composite group, the output of the command `symcg show CgName` identifies whether a group definition is stored in GNS. The sample output below shows that group `MyCompGroup` is stored in the GNS list since the output parameter **`CG in GNS`** has a value of **`Yes`**. It also shows the composite group's GNS mirror state. The group definition has been enabled for SRDF consistency using MSC.

```
% symcg show MyCompGroup

Composite Group Name: MyCompGroup

  Composite Group Type                              : RDF1
  Valid                                             : Yes
  CG in PowerPath                                   : No
  CG in GNS                                         : Yes (Is Mirror)
  RDF Consistency Protection Allowed                : Yes
  RDF Consistency Enabled                           : Yes

  Number of RDF (RA) Groups                         :    4
  Number of STD Devices                             :   64
  Number of BCVs (Locally-associated)              :    0
  Number of VDEVs (Locally-associated)             :    0
  Number of RBCVs (Remotely-associated STD-RDF)    :    0
  Number of BRBCVs (Remotely-associated BCV-RDF)   :    0
  Number of RRBCVs (Remotely-associated RBCV)      :    0

. . . .
```

GNS backup and restore mechanism

You can set the GNS daemon to automatically make a backup copy of local or global groups in a backup file. For global groups, the backup copy can be used to restore the global database on arrays. For local groups, the backup copy can be used to restore the groups when the primary group database file is corrupted. This backup file is *N* hours behind the in-memory database where *N* is a configurable backup interval with a default value of 6 hours.

Use the following daemon options to control the backup process:

To enable/disable backups — Set GNS_DB_BACKUP to either DISABLE or ENABLE where DISABLE is the default. When you set this option to ENABLE, the GNS daemon automatically backs up both the default local database and the global database into a backup file.

To set the backup interval time — Set the GNS_DB_BACKUP_INTERVAL to an integer value in hours. This option is only meaningful if GNS_DB_BACKUP option is set to ENABLE. The default backup interval is 6 hours. The valid range for the interval is (1..720).

All GNS database backup files are stored in the /var/symapi/gns directory. For both local and global GNS databases, up to two copies of backup files are maintained:

- For the GNS local database, the backup file names are local_group_db.backup and local_group_db.backup.bak, respectively.
- For the GNS global database, the backup file names are global_group_db.backup and global_group_db.backup.bak respectively.

Manual backups

You can manually backup to the GNS database by invoking the `stordaemon` program as follows.

```
stordaemon action storgnsd -cmd backup_db local

stordaemon action storgnsd -cmd backup_db global
```

Manual restores

You can manually restore from the GNS database by invoking the `stordaemon` program as follows.

```
stordaemon action storgnsd -cmd restore_db local

stordaemon action storgnsd -cmd restore_db global
changed_records:sid

stordaemon action storgnsd -cmd restore_db global all_records:sid
```

For global GNS database restore operations, there are two options, `changed_records` and `all_records`. The `changed_records` option only updates those records that differ from group records in the backup file. The `all_records` option rebuilds the entire database using the backup file as source data.

Troubleshooting GNS

When using GNS, be aware of the possibility of group name collision and invalid groups and what SYMAPI does to rectify each scenario.

Group name collisions

Each GNS daemon attempts to ensure that group names are unique (relative to a group type). It is possible, however, for duplicate-named groups to occur.

For example, it is possible for two different groups to be created with the same name on different arrays. A subsequent consideration change could then allow the GNS daemon to see both groups.

If a GNS daemon detects duplicate group names, it returns a state flag informing clients of this fact and modifies the group names to be unique by attaching a numeric suffix. For example, duplicate device groups with the name `MyDG` could be modified in the following way:

```
MyDG#12345678
MyDG#87654321
```

While in this state (duplicate names), the only operations permitted are those used to resolve the name conflict: group rename and delete (with the `-force` flag). For example:

```
symdg delete MyDG#12345678 -force
symdg rename MyDG#87654321 MyDG
```

Note: For information on renaming composite groups, see "Renaming a composite group" on page 133.

If users at two different hosts simultaneously create groups of the same type with the same name using devices on different arrays, both attempts might succeed. Because each host's GNS daemon is polling for changes made elsewhere, the daemons might not detect the name conflict until after the fact.

Invalid groups

A composite group is marked as invalid if one or more of the arrays that it resides on (in a GNS-enabled environment) cannot be reached. For example, each array that a consistency group spans records the identity of the other arrays that the group spans. If a discrepancy is noticed where a composite group is not defined on an array, even though some other array indicates it should be there, that group is assumed to be invalid.

Groups are also placed into an invalid state if certain internal bookkeeping information maintained by GNS is found to be missing or incorrect. For example, a consistency group spans two arrays and the GNS state recorded on each array verifies that the group also resides on the other array. If at some point one array is found to not contain the group (while the other array still thinks it should be found there), the group is assumed to be invalid.

This could happen if one of the arrays is reinitialized or a host attached only to one array decides to delete the composite group there. Since the group definition is also stored on an unreachable array, that group state is seen as invalid from that host. A user there would have to use the `-force` flag to delete the group. Groups marked as invalid can be deleted with the `-force` flag.

Displaying GNS data about groups

The GNS `stordaemon` list and show commands have been enhanced to support the following features.

List commands

- `stordaemon` *`action`* `storgnsd -cmd list_groups`
  Lists groups based on GNS daemon mode. If the GNS daemon is running with USE_GNS enabled, it will list all global groups. Otherwise, it will list all local groups in `storgnsd.db`.
- `stordaemon` *`action`* `storgnsd -cmd list_groups global`
  Lists all global groups.

- `stordaemon action storgnsd -cmd list_groups local`
  Lists all local groups.
- `stordaemon action storgnsd -cmd list_groups local:SYMCLI_DB_FILE`
  Allows you to list groups in a private GNS database determined by the environment variable SYMCLI_DB_FILE.

Show commands

- `stordaemon action storgnsd -cmd show_group cg:name`
  Shows composite groups based on GNS daemon mode. If the GNS daemon is running with USE_GNS enabled, it will list all global groups. Otherwise, it will list all local groups in `storgnsd.db`.
- `stordaemon action storgnsd -cmd show_group dg:name`
  Shows device groups based on GNS daemon mode. If the GNS daemon is running with USE_GNS enabled, it will list all global groups. Otherwise, it will list all local groups in `storgnsd.db`.
- `stordaemon action storgnsd -cmd show_group cg:name:local`
  Shows all local composite groups.
- `stordaemon action storgnsd -cmd show_group dg:name:local`
  Shows all local device groups.
- `stordaemon action storgnsd -cmd show_group cg:name:global`
  Shows all global composite groups.
- `stordaemon action storgnsd -cmd show_group dg:name:global`
  Shows all global device groups.
- `stordaemon action storgnsd -cmd show_group cg:name:local:SYMCLI_DB_FILE`
  Allows you to show composite groups in a private GNS database determined by the environment variable SYMCLI_DB_FILE.
- `stordaemon action storgnsd -cmd show_group dg:name:local:SYMCLI_DB_FILE`
  Allows you to show device groups in a private GNS database determined by the environment variable SYMCLI_DB_FILE.

CHAPTER 6
Fully Automated Storage Tiering

This chapter describes Fully Automated Storage Tiering™ (FAST) operations.

- Fully Automated Storage Tiering (HYPERMAX OS 5977)..... 156
- Fully Automated Storage Tiering (Enginuity 5874 to 5876)..... 166

Fully Automated Storage Tiering (HYPERMAX OS 5977)

Note: This section describes FAST operations for VMAX3 arrays running HYPERMAX OS 5977. "Fully Automated Storage Tiering (Enginuity 5874 to 5876)" on page 166 describes FAST for arrays running Enginuity levels 5874 through 5876.

The *EMC VMAX Family with HYPERMAX OS Product Guide* provides additional details about FAST concepts and operations.

EMC Fully Automated Storage Tiering (FAST) provides automated management of VMAX array disk resources on behalf of thin devices. FAST automatically configures disk groups to form a *Storage Resource Pool* by creating thin pools according to each individual disk technology, capacity and RAID type.

FAST technology moves the most active parts of your workloads (hot data) to high-performance flash disks and the least-frequently accessed storage (cold data) to lower-cost drives, leveraging the best performance and cost characteristics of each different drive type. FAST delivers higher performance using fewer drives to help reduce acquisition, power, cooling, and footprint costs. FAST is able to factor in the RAID protections to ensure write-heavy workloads go to RAID 1 and read-heavy workloads go to RAID 6. This process is entirely automated and requires no user intervention.

FAST further provides the ability to deliver variable performance levels through *Service Level Objectives*. Thin devices can be added to *Storage Groups* and the storage group can be assigned a specific Service Level Objective to set performance expectations.

A Service Level Objective is the response time target for your storage group. The Service Level Objective allows you to set the VMAX array with the desired response time target for your storage group. It automatically monitors and adapts to the workload in order to maintain (or meet) the response time target. The Service Level Objective includes an optional workload type so you can further tune expectations for the workload storage group to meet your performance objective.

There are five Service Level Objectives. For each one, except Optimized which does not allow you to set the I/O type, you can specify the workload type, such as Online Transaction Processing (OLTP) or Decision Support System (DSS). You cannot change a Service Level Objective but you can assign your storage group based on the service level required.

FAST monitors the storage groups performance relative to the Service Level Objective and automatically provisions the appropriate disk resources to maintain a consistent performance level.

VMAX3 arrays running HYPERMAX OS 5977 are custom-built and pre-configured with array-based software applications, including a factory pre-configuration for FAST that consists of:

- Data device (TDAT) — an internal device that is dedicated to providing physical storage used by thin devices.
- Data pool — a collection of data devices of identical emulation and protection type, all of which reside on disks of the same technology type and speed. The disks in a data pool are from the same disk group.
- Disk group — a collection of physical drives within the array that share the same performance characteristics.

- Storage Resource Pool — one (default) FAST Storage Resource Pool is pre-configured on the array. This process is automatic and requires no setup.

  You can list and display Storage Resource Pool configuration and generate reports detailing the demand that Storage Groups or Service Level Objectives are placing on the Storage Resource Pool.

The following sections describe how to perform tasks using SYMCLI commands:

- "Modifying Storage Resource Pools" on page 157
- "Reporting information" on page 158

Modifying Storage Resource Pools

You can use the `symconfigure set` command to modify the reserve capacity of an existing Storage Resource Pool and to set whether or not a Storage Resource Pool can be used for SRDF/A DSE operations.

Use the following syntax to modify a Storage Resource Pool:

```
set srp <SRP Name>,
     <[resv_cap = <n | NONE>]
      [,rdfa_dse = <ENABLE | DISABLE]>
```

- `resv_cap` is a percentage of the capacity of the Storage Resource Pool to be reserved for device write I/O activities. Valid values for the percentage are from 1 to 80. NONE disables it.

  For example, if you set the reserved capacity on a Storage Resource Pool to 30%, then the first 70% of the pool capacity is available for general purpose operations (host I/O allocations, local replication tracks and SRDF/A DSE allocations) and the final 30% of the pool capacity is reserved strictly for device write I/O activities.

  Note that existing TimeFinder snapshot sessions created on devices in the Storage Resource Pool will become invalid if the free capacity of the Storage Resource Pool, as a percentage of the usable capacity, goes below the reserved capacity.

- There is always one Storage Resource Pool assigned to be used for SRDF/A DSE allocations. The default Storage Resource Pool for FBA emulation is used by default. Enabling `rdfa_dse` on a Storage Resource Pool will make that pool be used for SRDF/A DSE allocations (and implicitly sets the currently assigned Storage Resource Pool to "disabled"). The maximum amount of storage from a Storage Resource Pool that can be used for DSE is controlled by the system wide `dse_max_cap` setting, as described in the *EMC Solutions Enabler SRDF Family CLI User Guide*.

For example, to set the existing Storage Resource Pool reserve capacity to a value of 10% and enable SRDF/A DSE operations, enter:

```
symconfigure -sid 230 commit -cmd "set srp Primary_SRP,
            resv_cap=10, rdfa_dse=ENABLE;"
```

Restrictions

- The following security privileges are required to execute this command:
  - Required Access type: CFGSYM
  - Required Authorization Rights: Storage Admin.
- The command will fail if the specified Storage Resource Pool does not exist.

- The command will fail if the Storage Resource Pool already has all its attributes set to the new values being specified.
- The command will fail if the target Storage Resource Pool for the operation is the default Storage Resource Pool for FBA emulation and the specified state is “disabled”.

Reporting information

You can list the Service Level Objectives, storage groups, and Storage Resource Pools configured on the array. You can also generate reports detailing the demand that storage groups or Service Level Objectives are placing on the Storage Resource Pools.

Listing Service Level Objectives

You can use the `symcfg` list and show commands to report Service Level Objectives with their available workloads.

For example, the following `symcfg list slo -detail` command lists all the Service Level Objectives configured on array 230:

```
symcfg -sid 230 list -slo -detail -all

                                SERVICE LEVEL OBJECTIVES

Symmetrix ID  : 00019710230

                     Approx
                     Resp
                     Time
Name       Workload  (ms)
---------  --------  -----
Optimized  N/A         N/A
Diamond    OLTP        0.7
Diamond    OLTP_REP    2.5
Diamond    DSS         2.5
Diamond    DSS_REP     4.5
Diamond    <none>      0.7
Platinum   OLTP        3.0
Platinum   OLTP_REP    4.5
Platinum   DSS         4.5
Platinum   DSS_REP     7.0
Platinum   <none>      3.0
. . .
Bronze     <none>     14.0
```

Note: If you issue the command without the `-all` qualifier, the output displays only the Service Level Objectives that can be set for your VMAX array configuration.

The `symcfg list -slo -v` command provides a verbose list of Service Level Objectives and workloads with additional descriptions.

Show a specific Service Level Objective

You can use the `symcfg show -slo` command to display a selected Service Level Objective. For example, the following command shows information about the Platinum Service Level Objective for array 230.

```
symcfg -sid 230 show -slo Platinum

Symmetrix ID  : 000197100230

Name          : Platinum

Workloads (5)
  {
                Approx
                Resp
                Time
   Name         (ms)
   --------     -----
   OLTP           3.0
   OLTP_REP       4.5
   DSS            4.5
   DSS_REP        7.0
   <none>         3.0
  }
```

Listing storage groups

Note: "Additional usage restrictions for storage groups are described in the following sections:" on page 93 explains how to create a storage group and how to add and remove devices from the group.

You can use the `symsg list -detail` command to show details for all the storage groups. The Service Level Objective, the workload, the RDF coordination setting and the Storage Resource Pool configured on a storage group are displayed.

You can sort information by Service Level Objective or by Storage Resource Pool by issuing `symsg list -detail -by_slo` or `symsg list -detail -by_srp`. You can also issue the `symsg list -v` option to display more information for all the storage groups.

For example:

```
symsg -sid 230 list -detail

S T O R A G E   G R O U P S


Symmetrix ID:        000197100230

                       Flags Number   Child FAST
Storage Group Name     EFMSL Devices  SGs    R   SLO Name   Workload SRP Name
------------------------------------------- ---- ---------- -------- --------------------
AcctPay                FX...     250      0  X   Gold       OLTP     Primary_SRP
Customer               AX...      38      0  .   Optimized  <none>   Primary_SRP
Ordering               MX...      19      0  X   Diamond    OLTP_REP Primary_SRP
Payroll                FX...      28      0  .   <none>     <none>   Secondary_SRP
TestOrdering           FX...      10      0  .   Bronze     <none>   <none>
testsg                 F....      10      0  .   <none>     <none>   <none>

Legend:
  Flags:
    Device (E)mulation  A = As400, F = FBA, 8 = CKD3380,
                        9 = CKD3390, M = Mixed, . = N/A
```

```
    (F)ast              X = Fast Managed, . = N/A
    (M)asking View      X = Contained in Mask View(s),   . = N/A
   Cascade (S)tatus     P = Parent SG, C = Child  SG,    . = N/A
   Host IO (L)imit      D = Defined,  S = Shared, B = Both, . = N/A
  Fast:
    (R)DF Coordination  X = Enabled, . = Disabled
```

When the (F)ast flag is set on an array running HYPERMAX OS 5977, it indicates that the storage group has a Service Level Objective (SLO) or a Storage Resource Pool (SRP) set on it and is FAST-managed.

The SLO Name, Workload, and SRP Name columns display the names of the Service Level Objective, workload and the Storage Resource Pool set on the storage group.

- A FAST-managed storage group that reports `<none>` for the SLO Name uses the Optimized Service Level Objective.
- A FAST-managed storage group that reports `<none>` for the SRP Name uses the array's default Storage Resource Pool for the emulation.

Listing Storage Resource Pools

You can use the `symcfg list -srp` command to display all the Storage Resource Pools configured on the array. For each Storage Resource Pool, the display lists its storage capacity and indicates whether it is used as the default for FBA devices and if it can be used for SRDF/A DSE operations. For example:

```
symcfg -sid 230 list -srp -GB

                          STORAGE RESOURCE POOLS

Symmetrix ID  : 00019710230

                                                   C A P A C I T Y
------------------------------- --- -------------------------------------
                                Flg     Usable  Allocated       Free Subs
Name                            DR        (GB)       (GB)       (GB)  (%)
------------------------------- --- ---------- ---------- ---------- ----
Primary_SRP                     FX     78848.0    42880.0    35968.0  125
Secondary_SRP                   ..     16360.0    10360.0     6000.0   75
                                    ---------- ---------- ---------- ----
Total                                  95208.0    53240.0    41968.0  116

Legend:
  Flags:
   (D)efault SRP : F = FBA Default, . = N/A
   (R)DFA DSE    : X = Usable, . = Not Used
```

Where:

- Usable — indicates the usable capacity of all the disk groups in the Storage Resource Pool.
- Allocated — indicates the sum of the device allocations, snapshot allocations, and SRDF/A DSE allocations on the Storage Resource Pool.
- Free — lists the difference between the usable and the allocated capacity.
- Subs % — indicates the percentage of the configured sizes of all the thin devices subscribed against the Storage Resource Pool.

Note: The capacity values are listed in gigabytes by default. You can display the capacity in megabytes or terabytes by specifying `-MB or -TB`.

You can use the `symcfg list -srp -v` command to display a verbose list of all the Storage Resource Pools in the array. For example:

```
symcfg -sid 230 list -srp -v -gb

Symmetrix ID             : 000197100230

Name                     : Primary_SRP
Description              : This is the system defined primary SRP
Default SRP              : FBA
Usable Capacity (GB)     : 78848.0
Allocated Capacity (GB)  : 42880.0
Free Capacity (GB)       : 35968.0
Subscribed Capacity (GB) : 98745.0
Subscribed Capacity (%)  :   125
Reserved Capacity (%)    :    10
Usable by RDFA DSE       : Yes

Disk Groups (4):
  {
   ---------------------------------------------
                                          Usable
                                 Speed   Capacity
   #   Name                  Tech (rpm)       (GB)
   --- -------------------- ---- ----- ----------
     1 DiskGroup_1          SATA  7200    20480.0
     2 DiskGroup_2          FC   10000     6144.0
     3 MyFCDiskGroup        FC   15000    51200.0
     4 MyEFDDiskGroup       EFD    N/A     1024.0
                                        ----------
   Total                                   78848.0
  }

Available SLOs (6):
  {
    Optimized
    Diamond
    Platinum
    Gold
    Silver
    Bronze
  }


Name                     : Secondary_SRP
Description              : This is another defined SRP
Default SRP              : None
Usable Capacity (GB)     : 16360.0
Allocated Capacity (GB)  : 10360.0
Free Capacity (GB)       :  6000.0
Subscribed Capacity (GB) : 12360.0
Subscribed Capacity (%)  :    75
Reserved Capacity (%)    :    10
Usable by RDFA DSE       : No

Disk Groups (3):
  {
   ---------------------------------------------
                                          Usable
                                 Speed   Capacity
   #   Name                  Tech (rpm)       (GB)
   --- -------------------- ---- ----- ----------
     5 DiskGroup_5          EFD    N/A     1024.0
     6 DiskGroup_6          FC   10000     6144.0
     7 MyFCDiskGroup2       FC   15000     8192.0
                                        ----------
```

```
   Total                                  15360.0
  }
Available SLOs (5):
  {
    Optimized
    Diamond
    Platinum
    Gold
    Silver
  }
```

The following information is listed for each Storage Resource Pool:

- The Default SRP field, which indicates whether the Storage Resource Pool is the default pool for FBA devices.
- Usable Capacity, which is the sum of the usable capacity of all the disk groups in the Storage Resource Pool.
- Allocated Capacity, which is the sum of the device allocations, the snapshot allocations and the SRDF DSE allocations on the Storage Resource Pool.
- Free Capacity, which is the difference between the usable and the allocated capacity.
- Subscribed Capacity, which is the sum of the sizes of all the thin devices subscribed against the Storage Resource Pool.
- Subscribed Capacity percentage, which is the percentage of the Usable Capacity that is used by thin devices.
- Reserved Capacity, which is a percentage of the Usable Capacity that is reserved for non-snapshot activities.

  Note: Existing TimeFinder snapshot sessions created on devices in the Storage Resource Pool can go invalid if the Free Capacity of the Storage Resource Pool, as a percentage of the Usable Capacity, goes below the Reserved Capacity.

- Whether the Storage Resource Pool can be used by SRDF/A DSE operations.
- The disk groups configured to the Storage Resource Pool. The list of disk groups is sorted by disk group number.
- The Service Level Objectives that are available for this Storage Resource Pool. The Optimized level is shown first and then the list is sorted from highest to lowest overall performance.

Showing a specific Storage Resource Pool

You can use the `symcfg show -srp` command to display a selected Storage Resource Pool. For example, the following command (with the detail option) shows detailed information about the pool named Primary_SRP.

```
symcfg -sid 230 show -srp Primary_SRP -detail -gb

Symmetrix ID             : 00019710230

Name                     : Primary_SRP
Description              : This is the system defined primary SRP
Default SRP              : FBA
Usable Capacity (GB)     : 78848.0
Allocated Capacity (GB)  : 42880.0
```

```
Free Capacity (GB)       : 35968.0
Subscribed Capacity (GB) : 98745.0
Subscribed Capacity (%)  :   125
Reserved Capacity (%)    :    10
Usable by RDFA DSE       : Yes
SRDF DSE Allocated (GB)  :     4.0

Disk Groups (4):
  {
   ---------------------------------------------
                                          Usable
                                   Speed   Capacity
   #   Name                  Tech (rpm)       (GB)
   --- -------------------- ---- ----- ----------
     1 DiskGroup_1          SATA  7200    20480.0
     2 DiskGroup_2          FC   10000     6144.0
     3 MyFCDiskGroup        FC   15000    51200.0
     4 MyEFDDiskGroup       EFD    N/A     1024.0
                                        ----------
   Total                                   78848.0
  }

Available SLOs (6):
  {
    Optimized
    Diamond
    Platinum
    Gold
    Silver
    Bronze
  }

Thin Devices(1200):
  {
   ------------------------------------------------
                        Thin Device        Snapshot
   Sym                Subs    Allocated    Allocated
   Dev   Emul         (GB)        (GB) (%)       (GB)
   ----- ----- ---------- -------------- ----------
   013E9 FBA       3000.0      102.0   4          0
   013EA AS400     3000.0      212.0   7        2.0
   013EB FBA       3000.0      112.0   4        2.0
    . . .
   00CEC FBA       2343.0      412.0  18        2.0
                ---------- ---------- --- ----------
   Total          98745.0    42800.0  44       76.0
  }
```

Reporting Storage Resource Pool demand

You can use the `symcfg list -srp -demand` command to report the subscribed and the allocated demand that is placed on the Storage Resource Pools.

Demand from Service Level Objectives

Specify option `-type slo` to report the demand placed by Service Level Objectives on the Storage Resource Pools. Devices that are not in a FAST managed storage group or are in a FAST managed storage group without an explicit Service Level Objective set will be reported against the Optimized Service Level Objective.

For example, the following command (with the detail option) reports the demand the Service Level Objectives are placing on the Storage Resource Pools:

```
symcfg -sid 230 list -srp -demand -type slo -detail -GB

                        STORAGE RESOURCE POOLS

Symmetrix ID  : 00019710230

Name                      : Primary_SRP
Usable Capacity (GB)      : 78848.0
SRDF DSE Allocated (GB)   :     4.0
Snapshots Allocated (GB)  :    76.0

-----------------------------------------------
                      Subscribed       Allocated
SLO Name  Workload    (GB)    (%)     (GB)   (%)
--------- -------- -------------- --------------
Optimized N/A          4000.0   5     1100.0  27
Platinum  OLTP        10034.0  12     5044.0  50
Gold      DSS_REP     64639.0  81    26873.0  41
Bronze    <none>      20072.0  25     9863.0  49
                   ---------- --- ---------- ---
Total                 98745.0 124    42880.0  43


Name                      : Secondary_SRP
Usable Capacity (GB)      : 16360.0
SRDF DSE Allocated (GB)   :     0.0
Snapshots Allocated (GB)  :    16.0

-----------------------------------------------
                      Subscribed       Allocated
SLO Name  Workload    (GB)    (%)     (GB)   (%)
--------- -------- -------------- --------------
Diamond   OLTP          512.0   3      512.0 100
Platinum  OLTP         1024.0   6      512.0  50
                   ---------- --- ---------- ---
Total                  1536.0   9     1024.0  67
```

Demand from by storage groups

Specify option `-type sg` to report the demand placed by storage groups on the Storage Resource Pools.

For example:

```
symcfg -sid 230 list -srp -demand -type sg -GB

                    STORAGE RESOURCE POOLS

Symmetrix ID             : 00019710230

Name                     : Primary_SRP
Usable Capacity (GB)     : 78848.0
SRDF DSE Allocated (GB)  :     4.0
Total Subscribed (%)     : 125

---------------------------------------------------------------------
                                                            Snapshot
                                Subscribed      Allocated  Allocated
SG Name                            (GB)        (GB)   (%)    (GB)
-------------------------------- ---------- -------------- ----------
<not_in_sg>                         1024.0      512.0  50        0.0
ee_sg230                            9216.0     4608.0  50       50.0
ee_sg230thin                       20480.0     0240.0  50       10.0
```

```
grs                                 68025.0    27520.0  40       16.0
                                 ----------  --------- --- ----------
Total                               98745.0    42880.0  43       76.0


Name                      : Secondary_SRP
Usable Capacity (GB)      : 16360.0
SRDF DSE Allocated (GB)   :     0.0
Total Subscribed %        :    75.0

-------------------------------------------------------------------
                                                            Snapshot
                                 Subscribed      Allocated  Allocated
SG Name                             (GB)        (GB)   (%)    (GB)
-------------------------------- ---------- -------------- ----------
<not_in_sg>                             0.0        0.0   0        0.0
rgi                                  5120.0     4608.0  90        0.0
sgc                                  6216.0     4728.0  68        0.0
test_sg                              1024.0     1024.0 100        0.0
                                 ----------  --------- --- ----------
Total                               12360.0    10360.0  83        0.0
```

Listing Storage Resource Pools for thin devices

You can use the `symcfg list -tdev -srp` command for all the thin devices in the array, to display the configured size, the Storage Resource Pools with which the devices are currently associated, a list of allocations for the devices, and their snapshots on each Storage Resource Pool. You can also provide a device range or a storage group name to report on a subset of the devices on the array. For example:

```
symcfg -sid 584 list -tdev -devs 13e9:1400 -srp

Symmetrix ID: 000197100584

                S Y M M E T R I X   T H I N   D E V I C E S
-------------------------------------------------------------------------------
                   Thin Device       Snapshots
Sym   Flgs      Total     Allocated   Allocated
Dev    EC        (GB)       (GB) (%)       (GB) SRP Name
----- ---- ---------- -------------- ---------- --------------------------------
013E9  FX      3000.0      102.0   4          0 Primary_SRP
013EA  AX      3000.0      212.0   7        2.0 Primary_SRP
013EB  F.      3000.0       12.0   1        1.0 Primary_SRP
        X                  100.0   4        1.0 Secondary_SRP
 . . .
01CEC  FX      2343.0      412.0  18        2.0 Secondary_SRP
            ---------- ---------- --- ----------
 Total        98745.0    42800.0  44       76.0

 Legend:
   Flags:
    Device (E)mulation  A = AS400, F = FBA
    (C)urrent SRP       X = Device is currently associated with SRP, . = N/A
```

Note: A device only has allocations on more than one Storage Resource Pool if the device is in the process of moving its allocations to a target Storage Resource Pool. (For example, the pool specified on the storage group that contained the device was changed to a different Storage Resource Pool).

Fully Automated Storage Tiering (Enginuity 5874 to 5876)

Note: This section describes FAST operations for arrays running Enginuity 5874 to 5876. "Fully Automated Storage Tiering (HYPERMAX OS 5977)" on page 156 describes FAST for arrays running HYPERMAX OS 5977.

For arrays running Enginuity 5874 to 5876, Fully Automated Storage Tiering (FAST) analyzes user workloads and moves volumes from one performance tier to another to optimize performance and/or economy. The array uses sophisticated algorithms to monitor activity and recommend strategies for moving the most-used data to the fastest (and most expensive) storage, such as Enterprise Flash Drives (EFD), the least-used data to the slowest (and least expensive) storage, such as SATA, while maintaining the remaining data on Fibre Channel (FC) drives.

For Enginuity 5874 to 5876, there are two FAST products: FAST and FAST for Virtual Pools (FAST VP). Table 5 shows the differences.

Table 5 FAST version differences (Enginuity 5876 and lower)

FAST on VMAX 20K and 40K	FAST VP on VMAX 10K, 20K and 40K
Requires Enginuity 5874 and higher	Requires Enginuity 5875 and higher
Supports standard devices	Supports thin devices
Supports FBA and CKD device emulations	• Supports FBA device emulation • Enginuity 5876 supports thin CKD 3390 and thin IBM i 512-byte D910 devices • Enginuity 5876 Q2 2013 SR provides support of thin IBM i 512-byte D910 devices on VMAX 10K systems
Disk group provisioning (DP) tiers: contain disk groups	Virtual pool (VP) tiers: contain thin pools
DP modes: Auto Approve and User Approve	VP modes: Auto Approve or None
User visible data movement plans and history	No plans or history generated
Federated Tiered Storage (eDisks) not supported	Supports Federated Tiered Storage (eDisks) with Enginuity 5876
FAST VP compression not supported	Supports FAST VP compression with Enginuity 5876 Q4 2012 SR and Solutions Enabler V7.5 and higher
Thin device/thin pool compression not supported	Supports compression for thin devices and thin (VP) pools with Enginuity 5876 Q4 2012 SR and Solutions Enabler V7.5 and higher
Three tiers per policy supported	Four tiers per policy supported with Enginuity 5876 Q4 2012 SR and Solutions Enabler V7.5 and higher

Systems that use FAST can:

- Reduce acquisition costs by maximizing utilization of less expensive drives (SATA) for infrequently used data
- Improve performance by optimizing data placement for high access requirements data
- Reduce overall operating costs and simplify management by consolidating systems, using fewer drives, and reducing power, cooling, and floor space requirements
- Reduce the total cost of ownership by optimizing the performance of a tiered system.

FAST configuration

FAST is available on arrays with Flash, Fibre Channel, SAS, or SATA drives and is supported on arrays with two or three of these drive types. To configure an array for FAST, the following actions are taken:

- Define VMAX array tiers — Tier configuration rules and restrictions are explained in "Managing tiers" on page 168.
- Define a FAST policy — Policy configuration rules and restrictions are explained in "Managing policies" on page 176.
- Define storage groups — Storage group configuration rules and restrictions are explained in "Managing storage groups" on page 181.

Configuration considerations

The following configuration considerations apply to arrays that use FAST software:

- Symmetrix Optimizer is required.
- FAST systems must configure Dynamic Reallocation Volumes (DRVs) to ensure sufficient space during swap operations. Optimizer uses DRVs to hold data internally during the data swap. Systems without DRV devices that have insufficient free space are degraded, and may be unable to complete a data swap.

  Arrays that use FAST software must follow the same DRV configuration requirements as Optimizer.
- FAST operations are not supported on thin devices.
- Time windows define when the FAST controller should collect performance information or execute data movement. Time windows are explained in "Setting time windows" on page 526.
- Control parameters define the numbers of and types of devices, the mode of operation, the thresholds for data movement, and the analysis time period. The control parameters are explained in "Managing the FAST controller" on page 186.

In addition, other tools are available to monitor your FAST configuration, including the following:

- Reports: The *compliance report* shows if the current placement of the devices in the storage group complies with the policy definition. The *technology demand report* shows the demand on each tier and technology, as per the current FAST configuration. These reports are explained in "FAST reports" on page 197.
- Many `list` and `show` commands are available to examine the details of the FAST configuration. These displays are shown in "Displaying FAST information" on page 196 and "FAST Output Examples" on page 437.

The SYMCLI commands for FAST are supported only on arrays that are attached locally to the host.

Note: To use FAST, the license SYMM_VMAX_FAST_TIERING must be enabled. The *EMC Solutions Enabler Installation Guide* provides all the licensing information.

The *EMC VMAX Family with Enginuity Product Guide* provides details on FAST and FAST VP concepts and sample configurations. The remainder of this chapter describes how to use the SYMCLI `symfast` and `symtier` commands for arrays running Enginuity levels 5874 through 5876.

Managing tiers

A VMAX array *tier* is the specification of the type of disk technology and speed (EFD, FC, SATA, or eDisk), the RAID protection type, and the specification of a set of resources (disk groups/virtual pools) from which the storage will be selected. Tiers can be created in the array for managing data performance with FAST, or as another way to group and report on the disk capacity usage.

A disk group provisioning (DP) tier is a set of disk groups with the same technology type. A DP tier has a target protection type. A virtual pool tier (VP) is a set of thin pools. A VP tier has a disk technology type and a RAID protection type. Members of a VP tier must be in thin pools with only DATA devices that reside on the tier technology type and match the tier protection type.

The `symtier` command provides the following capabilities for creating and managing tiers:

"Creating disk group tiers" on page 169

"Creating virtual pool tiers" on page 170

"Adding disk groups to a tier" on page 171

"Removing disk groups from a tier" on page 172

"Modifying a VP tier" on page 172

"Adding thin pools to a tier" on page 172

"Removing thin pools from a tier" on page 172

"Renaming tiers" on page 172

"Deleting tiers" on page 173

"Listing tier information" on page 173

"Showing tiers" on page 176

Tier restrictions

The following restrictions apply to tiers:

- VMAX array tier names cannot exceed 32 characters. Only alphanumeric characters, hyphens (-), and underscores (_) are allowed, however, the name cannot start with a hyphen or an underscore. Each tier name must be unique per array (across both disk group and virtual pool tiers), ignoring differences in case.
- The maximum number of tiers that can be defined on a VMAX array is 256, counting both DP and VP tiers. If creating a new tier exceeds this limit, then an existing tier must be deleted before creating the new tier.

- Disk groups can only be specified when the tier include type is static. When specifying the disk group in the command, the disk group technology type should match the tier technology. Both the disk group name and the disk group ID cannot be specified in the same command.
- The protection types 3+1, and 7+1 are only valid when used with `-tgt_raid5,` and protection type 6 + 2 and 14+2 are only valid when used with `-tgt_raid6`.
- An error returns on the `symtier create` action, if there are no matching disk groups to the tier technology.
- An error returns on the `symtier create` action, if a tier already exists with the exact same tier definition. Two tier definitions match when they have the same RAID type and disk groups.
- A DP tier cannot be created if it will partially overlap with an existing tier. Two tiers partially overlap when they share only a subset of disk groups. For example, TierA partially overlaps with TierB when TierA contains disk groups 1 & 2 and TierB contains only disk group 2. (Creating TierA will fail.)
- A mix of static and dynamic DP tier definitions is not allowed for a single technology type. However, you can have both static VP tiers and dynamic DP tiers of the same technology type. Likewise, you can have static VP and DP tiers of the same type.
- Dynamic VP tiers are not supported.

The following restrictions apply to virtual pools in VP tiers:

- A pool cannot be included in a VP tier if the pool is empty, disabled, or not a thin pool.
- The thin pool and thin tier technology type must match, meaning all DATA devices in the pool must reside on disks of the VP tier technology type.
- All DATA devices in the pool must reside on disks of the same storage class (disk technology type and speed), regardless of the SYMAPI_POOL_ALLOW_MIX_TYPE option.
- The protection type of the thin pool and VP tier must match.
- The emulation type of all thin pools included in a given VP tier must all be the same. FBA, CKD 3390, and IBM i 512-byte D910 device emulations are supported for DATA devices.
- A VP tier can contain a maximum of 4 thin pools.
- A thin pool cannot be included in more than one VP tier.

Creating disk group tiers

When creating a VMAX array tier for disk group provisioning, you must specify a tier name, the protection type (RAID 1, RAID 5, RAID 6), the technology (EDF, FC, SATA), and how the disk groups will be added to the tier. There are two ways disk groups can be added to a tier:

- Dynamic — A dynamic tier automatically includes all disk groups currently in the array that match the tier technology; this type of tier will expand to accommodate any newly added disk groups.

- Static — A static tier must be manually populated with disk groups. You can use disk group name(s) or disk group ID(s). Both disk group name and ID cannot be specified using the same command. If the *DiskGroupID* is not specified, an empty tier will be created.

The `-interval` and `-count` options can be used with the `symtier create` command to retry the command in case the SYMCLI fails to get an exclusive lock on the SYMAPI database.

When the FAST controller moves a device to a tier, it changes the device's protection type to match the tier's protection type.

Use the following form to create a tier:

```
symtier -sid SymmID [-i Interval] [-c Count]
   create -name TierName
          <-tgt_raid1 |
           -tgt_raid5 -tgt_prot <3+1 | 7+1> |
           -tgt_raid6 -tgt_prot <6+2 | 14+2>>
-technology <EFD | FC | SATA>
          -inc_type dynamic | [static
          [-dsk_grp <DiskGroupID[,DiskGroupID...] |
                     name:DiskGroupName[,DiskGroupName...]>]]
```

For example, to create a `static` tier on VMAX array `207` called `Primary` with `RAID 1 EFD` technology using disk group `1`, enter:

```
symtier -sid 207 create -name Primary -inc_type static -tgt_raid1 -technology EFD -dsk_grp 1
```

The new tier, `Primary`, now contains all the disks in disk group 1, resides on flash drives, and has RAID 1 protection. All disks in disk group 1 are now a part of this tier. When a storage group is associated with this tier through a FAST policy, the FAST controller may decide to move one or more devices to this tier. After the devices are moved to this tier, the devices will be in disk group 1 and their protection type will be RAID 1.

Creating virtual pool tiers

When creating a virtual pool (VP) tier, you must specify a tier name, the protection type (RAID 1, RAID 5, RAID 6), the technology (EDF, FC, SATA, external), and thin pools names. To be a member of a VP tier, a thin pool must contain only DATA devices that reside on the tier technology type and match the tier protection type.

Thin tiers can only be static, therefore the `-inc_type` option is not included in the syntax. To create a thin tier, use the following form:

```
symtier -sid SymmID [-i Interval] [-c Count]
create -name TierName
          <-tgt_unprotected |-tgt_raid1 |
           -tgt_raid5 -tgt_prot <3+1 | 7+1> |
           -tgt_raid6 -tgt_prot <6+2 | 14+2>>
          <-technology <EFD | FC | SATA> |
          [-external]
          -vp
          [-pool <PoolName[,PoolName...]>]
```

Where:

`-technology` — Specifies the technology type for the tier.

`-external` — Indicates the tier is external to the array.

Note: You can set the technology type on an FTS tier using both the `-external` and `-technology` options. SATA will be the default technology if a technology is not specified.

For example, to create a VP tier named `EFD_R5_VPTier` that has RAID 5 (3+1) protection and SATA disks on VMAX array 234, enter:

```
symtier -sid 234 create -name EFD_R5_VPTier -tgt_raid5 -tgt_prot 3+1
-technology SATA -vp
```

FTS tier restrictions

The following restrictions apply to creating external virtual tiers with Federated Tiered Storage (FTS):

- External tiers cannot be created with Enginuity versions lower than 5876.
- The `-tgt_unprotected` option is only valid for external tiers. In addition, it must be specified as the protection type for any external tier.
- External tiers (FTS) are only supported for FAST VP.
- If there are IBM i 512-byte D910 devices in the storage group, an external tier is not allowed as part of the associated policy.
- The optionally supplied VP pools must contain external-provisioned DATA devices when creating external tiers.
- The optionally supplied VP pools must not contain external-provisioned DATA devices when creating non-external tiers.

Adding disk groups to a tier

A disk group can only be added to a DP tier if the tier type is `static`. Use the following form to add disk groups to a tier:

```
symtier -sid SymmID [-i Interval] [-c Count]

  add -tier_name TierName [-propagate]
          -dsk_grp <DiskGroupID[,DiskGroupID...]> |
                   <name:DiskGroupName[,DiskGroupName...]>
```

Tiers in a VMAX array cannot partially overlap. (See the next example.) The `-propagate` option addresses the partial overlap restriction. When this option is used with the `add -disk_grp` command, the new disk group will also be added to all tiers that overlap with the tier being modified. This option only works when adding an unused disk group to a tier. That is, a disk group that does not participate in any tier.

For example, if there are two tiers, TierA with disk groups 1 and 2, and TierB with disk group 1 and 2, the following command can be issued:

```
symtier -sid 207 add -dsk_grp 3 -tier_name TierA -propogate
```

This command adds disk group 3 to TierA, TierB, and any other overlapping tiers.

Note: If TierA had only disk group 1, and TierB had only disk group 2, Solutions Enabler does not provide a way to make TierA contain disk group 1 and 2 and TierB contain disk group 1 and 2. The only way to do this is to delete the tiers and start again.

Removing disk groups from a tier

A disk group can be removed from a DP tier if the tier type is `static`. Use the following form to remove a disk group from a tier:

```
symtier -sid SymmID [-i Interval] [-c Count]
  remove -tier_name TierName [-propagate]
          -dsk_grp <DiskGroupID[,DiskGroupID...]> |
                   <name:DiskGroupName[,DiskGroupName...]>
```

For example, to remove disk group `2` from `PrimeTier`, enter:

```
symtier -sid 207 remove -dsk_grp 2 -tier_name PrimeTier
```

As with adding a disk group to a tier, you can use the `-propagate` option to remove the disk group from all tiers that overlap. The last disk group can not be removed from a tier that is part of a FAST policy.

Modifying a VP tier

The `modify` option has been added to the `symtier` command, as follows:

```
symtier -sid SymmID [-i Interval] [-c Count]
  modify -tier_name TierName
  -technology <EFD | FC | SATA>
```

Use the `-technology` option to specify the tier type when modifying an external tier.

Adding thin pools to a tier

Use the following form to add a thin pool to a VP tier:

```
symtier -sid SymmID [-i Interval] [-c Count]

add -tier_name TierName
          -pool <PoolName[,PoolName...]>
```

Note: An error returns if you try to add a thin pool that is already included the tier.

To add thin pool `EFD_R5_POOL` to the `EFD_R5_VPTier` on VMAX array 234, enter:

```
symtier -sid 234 add -pool EFD_R5_POOL -tier_name EFD_R5_VPTier
```

Removing thin pools from a tier

Use the following form to remove a thin pool from a VP tier:

```
symtier -sid SymmID [-i Interval] [-c Count]

   remove -tier_name TierName
          -pool <PoolName[,PoolName...]>
```

You cannot remove all thin pools from a VP tier if the tier is part of a FAST policy. In addition, you cannot remove a pool from a tier if the tier is in an associated policy, and a device in the storage group is bound to the pool.

Renaming tiers

A tier name can be changed using the `symtier rename` command, as follows:

```
symtier -sid SymmID [-i Interval] [-c Count]
   rename -tier_name TierName -name NewTierName
```

For example, to rename the tier `Primary` to `PrimeTier`, enter:

```
symtier -sid 207 rename -tier_name Primary -name PrimeTier
```

Any new name must adhere to all"Tier restrictions" on page 168.

Deleting tiers

Before deleting a tier, make sure it is not associated with any FAST policy. Use the following form to delete a tier:

```
symtier -sid SymmID [-i Interval] [-c Count]
   delete -tier_name TierName [-force]
```

The `-force` option is required to delete a non-empty static tier.

For example, to delete a tier named `DBTier`, enter:

```
symtier -sid 207 delete -tier_name DBTier
```

The `DBTier` is deleted from VMAX array 207.

Listing tier information

The `symtier list` command gives a summary of all the disk group or virtual pool tiers in the array. The `symtier show` command includes more details about each tier.

```
symtier [-sid SymmID] [-v] [-offline]
    list [-dp | -vp [-ckd] [-fba]]
         [-technology <EFD | FC | SATA> | -external]

symtier show -sid SymmID [-offline]
```

Note: Prior to using the `-offline` option, the SYMAPI database must be populated with the tier information by using the command `symcfg sync -fast`.

When listing the tier information (without the `-v` option), you can specify `-dp` (disk group provisioning) or `-vp` (virtual pool provisioning). When neither option is specified, all tiers display without their capacity information.

The `symtier list` display contains two columns, one for the emulation type of FAST VP tiers and one for disk location (internal or external). FAST tiers and empty FAST VP tiers will display N/A.

For example, to list all the tiers on VMAX array `432`, enter:

```
symtier list -sid 432

Symmetrix ID          : 000194900432

-------------------------------------------------------
                          L
                          o      Target             Flgs
Tier Name             Type c Tech Protection   Emul   ID
--------------------- ---- - ---- ------------ ----- ----

HR_DP                 DP   I SATA RAID-1       N/A    S-
HR_TEST_TIER          VP   I SATA RAID-1       FBA    S-
test_366859           VP   I SATA RAID-1       3390   S-
test_external         VP   X N/A  Unprotected  FBA    SX

Legend:
```

```
  Tier (Type)      :  DP = Disk Group Provisioning, VP = Virtual Pools
  Disk (Loc)ation  :  I = Internal, X = External
 Flgs:
  (I)nc Type       :  S = Static, D = Dynamic
```

When the `-dp` option is specified, only the disk group tiers display, including their capacity information. When the `-vp` option is specified, only the virtual pool tiers display, including their capacity information.

For example, to list all the thin tiers on VMAX array `432`, enter:

```
symtier list -vp -sid 432

Symmetrix ID          : 000194900432

-------------------------------------------------------------------------------
                      L                                Logical Capacities (GB)
                      o      Target             Flgs -------------------------
Tier Name             c Tech Protection   Emul   ID   Enabled     Free     Used
--------------------- - ---- ------------ ----- ---- -------- -------- --------

HR_TEST_TIER          I SATA RAID-1       FBA    S-         4        4        0
test_366859           I SATA RAID-1       3390   S-         0        0        0
test_external         X N/A  Unprotected  FBA    SX         4        4        0

Legend:
Disk (Loc)ation       :  I = Internal, X = External
 Flgs:
  (I)nc Type      :  S = Static, D = Dynamic
```

The columns in the output are defined as follows:

`Tier Name` — The name of the tier.

`Loc(ation)` — Internal or external tier.

`Tech` — Technology of the tier.

`Target Protection` — Tier protection type.

`Emul` — Device emulation type.

`Flgs ID` — Specifies whether the tier is static or dynamic and whether the tier is enabled for dynamic discovery.

`Enabled Capacity (Logical)` — Total pool enabled capacity for all thin pools in the tier.

`Free Capacity (Logical)` — Total enabled capacity minus the used capacity, (minimum 0).

`Used Capacity (Logical)` — Total pool allocated capacity for all thin pools in the tier. Allocated capacity on all DATA devices will be counted, including DATA devices that are not enabled; therefore Used may be greater than Enabled.

The next example lists the disk group tiers on VMAX array 234:

```
symtier -sid 234 list -dp

Symmetrix ID          : 000194900234

---------------------------------------------------------------------
                                        I  Logical      Raw  Logical
                           Target       n  Max Cfg Unconfig   Config
Tier Name             Tech Protection   c     (GB)     (GB)     (GB)
--------------------- ---- ------------ - -------- -------- --------
ArchiveTier           SATA RAID-5(7+1)  D    78464    88113     1365
PrimeTier             EFD  RAID-1       S      645      290      500
WorkTier              FC   RAID-5(3+1)  D     2753     1030     1980

Legend:
  Inc Type      :  S = Static, D = Dynamic
```

The columns unique to the disk group tier display follow:

`Logical Max Cfg (GB)` — An estimate of the potential maximum logical configured capacity for the tier, if all the unconfigured space in the tier disk groups was used to create one large device of the tier protection type. The tier raw unconfigured capacity is multiplied by a factor based on tier protection type to estimate how many more logical GB worth of devices could reside on the tier (multiplied by 7/8 for a RAID-5 (7+1) tier). The resulting estimate is added to the current tier logical configured capacity to get the maximum configurable capacity. If a disk group in the tier does not have enough usable disks to support devices of the tier protection type (disk count of 7 for a RAID-5 (7+1) tier), that disk group does not contribute any capacity towards the tier maximum configurable capacity.

This value is only an estimate, and it is not guaranteed that the full maximum configurable capacity can be reached for the tier, because the estimate does not account for:

- Affinity groups.
- The physical layout of hypers on the disks.
- If standard tiers overlap, the unconfigured space from disk groups included in both tiers will be attributed to each tier, therefore the same unconfigured space may be counted multiple times.

`Raw Unconfig (GB)` — The unconfigured space in the tier disk groups. If disk group tiers overlap, the unconfigured space from disk groups included in both tiers will be attributed to each tier, therefore the same free space may be counted multiple times. If a disk group does not have enough usable disks to support devices of the tier protection type (disk count of 7 for a RAID-5 (7+1) tier), raw free capacity for that disk group will be reported as 0.

`Logical Config (GB)` — The sum of the logical capacity of all devices that match the tier protection type and reside on the tier disk groups.

Verbose listing

Use the `-v` option to expand the output to include more details, as shown in the following command and output example:

Showing tiers

In the `show` display, a new `Tier Type` field displays. In addition, for thin tiers, a new `Thin Pools` table displays, as shown in the following example command and output:

```
symtier -sid 432 show -tier_name AS400_TIER

Symmetrix ID          : 000194900432

Tier Name             : AS400_TIER
Tier Type             : VP
Technology            : SATA
Target Protection     : RAID-1
Emulation             : FBA
Include Type          : Static

Thin Pools(1)
    {
    ---------------------------------------------------
                         Logical Capacities (GB)
                 Dev   -------------------------- Full
    Pool Name    Emul   Enabled     Free     Used  (%)
    ------------ ----- -------- -------- -------- ----
    AS400_TP     FBA          4        2        2   50
                       -------- -------- --------
    Total                     4        2        2
    }

Legend:
  Tier Type     :  DP = Disk Group Provisioning, VP = Virtual Pools
```

Managing policies

A FAST *policy* is a grouping of 1 to 3 tiers (or 1 to 4 tiers for VP FTS tiers) and an assigned upper limit of how much each associated storage group (application) or VP pool can use from the tier. When creating a FAST policy, the most important component that must be set is percentage of the storage group that can reside on each tier.

This following sections detail policy management operations.

Policy restrictions

The following restrictions apply to FAST policies:

- Policy names must be unique and cannot exceed 32 characters. Policy names are case insensitive, in other words, HRPolicy and hrpolicy are not unique names.
- Only alphanumeric characters, hyphens (-), and underscores (_) are allowed. The policy name cannot start with a hyphen or underscore.
- An array can have up to 256 policies.
- There can be a maximum of up to four VP tiers in a storage policy, and each tier must be unique, there can be no overlapping disk groups. For Solutions Enabler V7.4 and earlier versions, the maximum is three tiers in a storage policy.
- A policy cannot have an empty tier.

- A FAST policy may contain up to three disk group (DP) tiers or up to four virtual pool (VP) tiers, but not a combination of both DP and VP tiers. The first tier that is added to a given policy determines the type of tier that policy will contain. The fourth tier in a FAST VP policy can only be an external FTS tier.
- The emulation type of included thin pools must match across all VP tiers in a FAST policy. For example, A FAST policy cannot contain both VP tiers with FBA thin pools and VP tiers with CKD thin pools.
- If specified, the capacity limit for the tier (*MaxSgPercent*) must to be greater than 0 and less than or equal to 100. All other values will be rejected.

Creating a FAST policy

FAST policies are created using the `symfast` command, as follows:

```
symfast -fp -sid SymmID [-i Interval] [-c Count]
   create -name FastPolicyName [-tier_name TierName
         [-max_sg_percent MaxSgPercent]]
```

The `-emulation` option is only available for arrays running Enginuity 5876 and higher. If the emulation type is not provided, the default emulation is FBA. Mixed emulation VP types are not allowed in the same policy.

The *MaxSgPercent* specified in a FAST policy indicates the upper limit of how much of the total storage group's logical capacity can be allocated to a specific tier.

For example, the assigned capacity for Tier 1 is 40% and the storage group associated with the FAST policy has five devices, each of them 40 GB. So the total capacity is 200 GB and the space available in Tier 1 for this storage group is at most 80 GB. Consequently, the FAST controller will not place more than two devices on Tier 1.

Note: The `-max_sg_percent` option is no longer required. If it is not specified, the default value is 100.

The FAST controller uses this capacity percent as an upper limit and does not exceed the limit. In the above case if FAST policy had 50% capacity for Tier 1, that is 100 GB for Tier 1. The FAST controller would still only place at maximum two devices on the Tier 1 since adding one more device will exceed the upper limit of 50%.

To create a FAST policy, you must supply the VMAX array ID, a policy name, at least one tier name, and the upper limit (percent) of the tier that the storage group can occupy.

Note: The limit for the tier in the policy, `-MaxSgPercent`, will be computed using only the capacity of thin devices in the associated storage group; standard devices in the storage group are excluded from the calculation. Additionally, only allocated tracks from the thin devices will be counted, not logical device capacity.

For example, to create a policy with the name `DBPolicy`, add the tier `PrimeDBTier` to the policy, and specify a capacity of `30`% of a storage group, enter:

```
symfast -fp -sid 207 create -name DBPolicy -tier_name PrimeDBTier -max_sg_percent 30
```

As per the policy, no more than 30% of a storage group associated with this policy will reside on PrimeDBTier.

Adding a tier to a policy

The following restrictions apply when adding a tier to a policy:

- An empty tier cannot be added; the maximum tiers for each policy is three.
- A new tier cannot be added if it will result in a configuration where two tiers in the same policy share a common disk group.
- When adding a VP tier to a FAST policy, the policy must be empty or contain only VP tiers.
- A policy can only have the same emulation VP tiers.
- A given VP tier may be included in multiple FAST policies.

To add a tier to a policy, use the following form:

```
symfast -fp -sid SymmID [-i Interval] [-c Count]
            -fp_name FastPolicyName
  add -tier_name TierName
            -max_sg_percent MaxSgPercent
```

For example, to add the tier `ArchiveDBTier` to the policy `DBPolicy` and specify that the tier can contain up to `10`% of the storage group associated with this policy, enter:

```
symfast -fp -sid 207 add -tier_name ArchiveDBTier -max_sg_percent 10 -fp_name DBPolicy
```

For any storage group associated with the DBPolicy, no more than 10% of its storage should be in `ArchiveDBTier`.

Modifying the limit for a tier in a policy

To modify the limit of the tier capacity available to a storage group, use the following form:

```
symfast -fp -sid SymmID [-i Interval] [-c Count]
            -fp_name FastPolicyName
  modify -tier_name TierName
               -max_sg_percent MaxSgPercent
```

For example, to change the limit of `PrimeDBTier` to `70`% in `DBPolicy` on VMAX array `207`, enter:

```
symfast -fp -sid 207 modify -fpname DBPolicy-tier_name PrimeDBTier -max_sg_percent 70
```

As per the policy, no more than `70`% of the storage group associated with this policy will reside on `PrimeDBTier`. If the policy is associated, the sum of the limits must still be equal to or greater than 100%.

Removing a tier from a policy

Before removing any tier from a policy, make sure the `remove` action will not result in the sum of the tier limits being less than 100%. If the tier is the only tier in the policy, and the policy is associated with a storage group, the tier cannot be removed. A VP tier cannot be removed from a policy if thin devices in the associated storage group are bound to pools in the tier.

Use the following form to remove a tier from a policy:

```
symfast -fp -sid SymmID [-i Interval] [-c Count]
            -fp_name FastPolicyName
   remove -tier_name TierName
```

For example, to remove `ArchiveTier` from the `FinanceData` policy on VMAX array `207`, enter:

```
symfast -fp -sid 207 -fp_name FinanceData remove -tier_name ArchiveTier
```

The `ArchiveTier` is removed from the policy.

Renaming a policy

When renaming a policy, make sure the new name does not already exist. To rename a FAST policy, use the following form:

```
symfast -fp -sid SymmID [-i Interval] [-c Count]
            -fp_name FastPolicyName
   rename -name NewFastPolicyName
```

For example, to rename `DBPolicy` to `OraDBPolicy` on VMAX array `207`, enter:

```
symfast -fp -sid 207 rename -fp_name DBPolicy -name OraDBPolicy
```

The new name for the policy is `OraDBPolicy`.

Deleting a policy

The following restrictions apply to deleting FAST policies:

- A policy that has tiers can only be deleted by using the `-force` flag.
- The policy must not be associated with any storage group. If the policy is associated, attempts to delete the policy return an error. To delete the policy, first disassociate the storage group from the policy, and then delete the policy.

To delete a FAST policy, use the following form:

```
symfast -fp -sid SymmID [-i Interval] [-c Count]
            -fp_name FastPolicyName
delete -fp_name FastPolicyName [-force]
```

For example, to delete FAST policy `HRPolicy` on VMAX array `207`, enter:

```
symfast -fp -sid 207 delete -fp_name HRPolicy
```

The FAST policy `HRPolicy` is deleted from the array.

Listing and showing policies

To list and show FAST policies, use the following form:

```
symfast [-sid SymmID] [-offline] [-v]
   list -fp [-v] [-dp | -vp [-ckd | -fba]]

symfast [-sid SymmID] [-offline]
   show -fp_name FastPolicyName
```

Notice that the `list` command includes an optional filter for disk group provisioning (`-dp`) or virtual pools (`-vp`) and the emulation type (`-ckd`) and (`-fba`).

Note: Prior to using the `-offline` option, the SYMAPI database must be populated with the FAST information by using the command `symcfg sync -fast`.

For example, in the policy output, when you specify the `-dp` option, only the policies containing disk group tiers display. When you specify the `-vp` option, only the policies containing virtual pool tiers display. When neither option is specified, all policies display.

To show the FAST policies for VMAX array `601` that contain VP tiers, enter:

```
symfast -sid 601 show -fp_name test_policy

Symmetrix ID          : 000195700601

Policy Name           : test_policy
Emulation             : FBA

Tiers(4)
    {
    ---------------------------------------------------------------------
                                                   L
                                            Max SG O       Target
    Tier Name                        Type  Percent C Tech  Protection
------------------------------- ---- -------- - ----- -------------
    test_EFD                         VP         10 I EFD   RAID-1
    test_FC                          VP         30 I FC    RAID-5(3+1)
    test_SATA                        VP         40 I SATA  RAID-6(6+2)
    test_FTS                         VP         70 X N/A   Unprotected
    }

Storage Groups(1)
    {
    ------------------------------------
    Storage Group Name               Pri
    -------------------------------- ---
    test_sg                          1
    }

Legend:
  Tier Type       :  DP = Disk Group Provisioning, VP = Virtual Pools
  Disk (Loc)ation :  I = Internal, X = External
```

For example, when a FAST VP policy is listed with the `-v` option, a flags (`Flgs`) column indicates if the tier can be compressed, as follows:

```
symfast -sid 601 list -fp -v

Symmetrix ID          : 000195700601

Policy Name           : test_fp
Emulation             : FBA

Tiers(1)
    {
    ------------------------------------------------------------------------
                                                   L
                                            Max SG O       Target        Flgs
    Tier Name                        Type  Percent C Tech  Protection     C
    -------------------------------- ---- -------- - ----- ------------- ----
    test_tier                        VP        100 I FC    RAID-1         .
    }

Storage Groups(1)
    {
    ------------------------------------
    Storage Group Name               Pri
    -------------------------------- ---
    test_fast                        2
```

```
    }

Legend:
  Tier Type        :  DP = Disk Group Provisioning, VP = Virtual Pools
  Disk (Loc)ation  :  I = Internal, X = External

 Flgs:
  (C)ompression    :  X = Compression Capable, . = Not Compression Capable
```

Managing storage groups

Storage groups are a collection of devices on the VMAX array that can be used for masking/mapping, Virtual LUN Technology, and FAST operations. This section explains the FAST-related storage group options.

Note: "Adding cascaded storage groups (Enginuity 5876 and HYPERMAX OS 5977 only)" on page 101 explains how to create a storage group and how to add and remove devices from a group.

The following sections detail storage group management operations.

FAST storage group and device restrictions

The following restrictions apply to storage groups that are associated with FAST policies:

- A storage group can be associated with only one policy. The policy must have at least one tier.
- FAST policies may only be associated to storage groups containing devices. A parent storage group containing other storage groups cannot be associated to a FAST policy.
- The emulation type of all devices in the storage group, including both standard and thin devices, must be the same.
- Thin device emulation types supported by FAST VP are FBA, CKD 3390, and IBM i 512-byte D910 devices.
- The array can have a maximum of 8192 storage group associations to policies. Further, only 1000 of these associations can be with policies containing VP tiers.
- The total of the limits in the FAST policy must add up to at least 100%.
- A device can be associated with only one policy. An attempt to associate a storage group with a device that is already associated to a policy will be blocked. This restriction applies to all devices, standard and thin, associated with any FAST policy, regardless of whether the policy contains DP or VP tiers. For example, a standard device in a storage group associated with a policy containing VP tiers could not be added to a different storage group associated with a policy containing DP tiers. To place all of the standard and thin devices under FAST control, two separate storage groups would have to be created.
- Only the metahead is allowed in the storage group; metamembers cannot be part of a storage group.
- Storage groups containing a mix of standard and thin devices can be associated with a FAST policy containing VP tiers, however, FAST will only operate on the thin devices, and only the capacity from thin devices will be used to determine compliance.

Conversely, when the FAST policy contains DP tiers, FAST will only operate on the standard devices, and only the capacity from standard devices will be used to determine compliance.

All devices in the storage group are considered under FAST control, however, a choice must be made whether FAST will manage the thin devices or the standard devices in a given storage group.

- Devices that are not movable cannot be added to a storage group that is (or will be) associated with a FAST policy. FAST does not support moving these device types:
  - AS400, ICOS, ICL
  - CKD EAV, CKD EAV phase 3, and CKD concatenated metadevices
  - Diskless
  - DRV
  - SAVE
  - SFS
  - Unprotected
  - VDEV (can be added to a storage group, but will be ignored for FAST operations)
- If the FAST policy contains VP tiers, none of the thin devices can be bound to a pool outside of the policy.
- The FAST policy can contain multiple VP tiers with the same technology type, however FAST VP only performs *compliance* movements between such tiers. There will be no *performance* movements between tiers of the same technology type within a single FAST policy.
- No checks are made for oversubscription at association time. A virtual pool in a VP tier in an associated FAST policy becomes a shared resource; its capacity is no longer used not only by the bound thin devices but also thin devices under FAST control. Therefore, even a pool that is not oversubscribed (`MAX_SUBS_PERCENT <= 100%`) may become full due to a combination of allocations for bound devices and FAST moving chunks into the pool.

Associating a storage group to a policy

Note: Setting the `-priority` on an association is optional. If a priority is not provided, a default priority of 2 will be automatically set.

Use the `symfast` command to associate a storage group to an existing FAST policy. This operation associates a storage group to an existing FAST policy and assigns a priority value for the association. The priority of a storage group affects which storage group is serviced first by the tier when there is contention among FAST storage groups for space in a storage tier. The values are 1, 2, or 3. Priority 1 has the highest value, priority 3 has the lowest value.

Setting SRDF coordination

Previously, when R1 devices and R2 devices were both managed by FAST, there was no coordination of FAST VP actions (moves) between the local and remote VMAX arrays. Both sets of devices were managed individually, based on statistics collected locally on each array. Without coordination, the R1 device and the corresponding R2 device could reside in different tiers, which could impact RDF performance.

Solutions Enabler running Enginuity V5876 provides an option to set FAST VP RDF coordination on an association. This instructs FAST to factor the R1 device statistics into the move decisions that are made on the R2 device. This attribute can be set on a storage group, even when there are no SRDF devices in the storage group.

To associate a storage group with a FAST policy, and set any associated attributes, use the `symfast` command, as follows:

```
symfast -fp -sid SymmID [-i Interval] [-c Count]
            -fp_name FastPolicyName
   associate -sg SgName [-priority PriorityValue ]
```

Setting priority on an association is optional. Solutions Enabler sets the default priority at 2 if a value is not supplied.

For the SRDF coordination, both the R1 and the R2 devices must reside on an array running Enginuity 5876.

You cannot associate a parent storage group with a FAST policy.

For example, to associate storage group `OraSales` with the FAST policy `DBPolicy`, and set the storage group priority at `1`, enter:

```
symfast -sid 207 associate -sg OraSales -fp_name DBPolicy -priority 1
```

This association is successful when the storage group configuration rules have been followed and there is enough capacity allocated in the tiers that belong to the FAST policy. The association will not be blocked if the tiers do not have sufficient capacity, but the storage group will be out of compliance with its policy.

After the storage group is associated with the policy, the FAST controller will actively monitor the storage group and suggest or run optimizations to improve performance.

Reassociating a storage group to a new policy

You can reassociate a storage group to a new policy. When a storage group is reassociated, all the current attributes set on the original association automatically propagate to the new association. This feature eliminates the previous process of disassociating a storage group, then associating the group to a new policy, and entering the attributes, such as priority, on the association.

The `reassociate` option has been added to the `symfast` command, as follows:

```
symfast -sid SymmID [-i Interval] [-c Count]
            -sg_name SgName
reassociate -fp_name FastPolicyName
```

Restrictions

The following restrictions apply to the reassociate action:

- The storage group name must be a valid storage group name.
- The storage group and policy must already exist on the array.
- The storage group must be in an association before performing a reassociation.
- The new policy for the storage group, must have the same emulation as the storage group. Mix emulation association will result in an error.

- The storage group cannot be associated with empty policy, the reassociated policy must contain at least one tier.
- The total of the capacity percentage for the target FAST policy must add up to at least 100%.
- If the FAST policy contains VP Tiers, all of the thin devices in the storage group must be bound to any VP pool in a tier in the policy. None of the thin devices can be bound to a pool outside of the policy.

Disassociating a storage group from a policy

When a storage group is disassociated from a FAST policy, the FAST controller no longer tries to optimize the performance of that storage group. To disassociate a storage group from a policy, use the following form:

```
symfast -fp -sid SymmID [-i Interval] [-c Count]
        -fp_name FastPolicyName
      disassociate -sg SgName
```

For example, to disassociate storage group `OraSales` from the FAST policy `DBPolicy`, enter:

```
symfast -sid 207 disassociate -sg OraSales -fp_name DBPolicy
```

The disassociate operation occurs, even if the FAST controller is currently moving devices in the storage group.

Disassociating a storage group containing thin devices means those devices in the storage group are no longer under FAST control. Data from the thin devices remains where it is, even if it is spread across multiple pools. FAST will not move the data back to the bound pool.

Modifying a storage group

With Enginuity 5876, you can modify a storage group priority and the SRDF coordination. The SRDF coordination is only applicable to FAST VP.

The priority of the storage group can only be changed if the storage group is associated with a FAST policy.

To change a storage group priority or the SRDF coordination attribute, use the `symfast modify` command, as follows:

```
symfast -sid SymmID [-i Interval] [-c Count]
         -fp_name FastPolicyName
 modify -sg SgName [-priority PriorityValue]
```

The priority values can be 1 (highest), 2, or 3 (lowest). The SRDF coordination attribute is only applicable for FAST VP.

For example, to decrease the priority of storage group `OraMarketing`, associated with FAST policy `DBPolicy`, from `1` to `2`, enter:

```
symfast -sid 207 modify -sg OraMarketing -fp_name HRDBPolicy -priority 2
```

The storage group `OraMarketing` has a priority of `2`.

Displaying storage group associations

Use the `symfast list -association` command to display the storage groups, their priority, their associated policies, and whether the SRDF coordination attribute is set.

Note: Before using the `-offline` option, the SYMAPI database must be populated with the storage group, tier, and FAST data using the `symcfg sync -fast` command.

For example, to display the storage group associations on VMAX array `432`, enter:

```
symfast list -sid 432 -association
```

Output similar to the following displays:

```
Symmetrix ID          : 000194900432

-------------------------------------------------------------------------
Storage Group Name               Policy Name                      Pri Flgs
                                                                       R
-------------------------------- -------------------------------- --- ----
HR_SG                            HR_FP                            3    X
HR_TEST                          HR_FP                            2    .

Legend:
 Flgs:
   (R)DF Coordination : X = Enabled, . = Disabled
```

The output is sorted alphabetically by storage group name.

The `show` action of `symfast` displays information about associations between FAST policies containing standard or thin tiers and storage groups.

```
symfast [-sid SymmID] [-offline]
    show -association -sg SgName [-all] [-v]
```

The show association display includes a `Tier Type` column in the `Tiers` table. A new table lists thin devices. By default, only the devices managed by FAST display (thin devices in the case of a policy containing thin tiers, and standard devices in the case of a policy containing standard and thin tiers); gatekeeper devices are filtered out.

Use the `-all` option to show all devices in the storage group, regardless of the associated policy.

The `symfast show -association` output shows FAST VP associations as Enable or Disable. Disable displays for all FAST VP associations on arrays running versions lower than Enginuity 5876. FAST (DP) associations will display N/A.

For example, to show the associations for storage group `hr_sg` on VMAX array `432`, enter:

```
symfast show -sid 432 -association -sg hr_sg
```

```
Symmetrix ID          : 000194900432

Storage Group         : hr_sg

Thin Devices(4)
    {
    ---------------------------------------------------
          Flgs Dev        Total Bound        Allocated
    Sym    PC  Emul      Tracks Pool Name       Tracks
    ----- ---- ----- ---------- ------------ ---------
    0062A  N.  FBA        33000 HR                  12
    0062B  N.  FBA        33000 N/A                  -
```

```
    0062C  N.  FBA        33000 N/A                  -
    0062D  N.  FBA        33000 HR               32796

    Total           ----------              ---------
    Tracks               66000                   32808
       GBs                   4                       2
    }

Policy Name          : hr_fp
Priority             : 3
RDF Coordination     : Disabled

Tiers(1)
    {
    ---------------------------------------------------------------------------
                                                   L
                                            Max SG O       Target        Flgs
    Tier Name                       Type  Percent C Tech  Protection     C
    ------------------------------- ---- -------- - ----- ------------- ----
    hr_tier                         VP        100 I SATA  RAID-1
    }

Legend:
 Tier Type: DP = Disk Group Provisioning, VP = Virtual Pools
 Device Flags:
   (P)inned      : Y = Device is Pinned, N = Device is not Pinned
   (C)ompression : X = Compressed Allocation, . = No Compressed Allocation
 Tier Flags:
   (C)ompression : X = Compression Capable, . = Not Compression Capable
```

Note: The order of the associated tiers displays highest to lowest, meaning the highest performing tier is listed on top and the lowest performing tier is listed on the bottom.

A verbose `show association` display can be generated. In addition to the information shown by the non-verbose display, the verbose display shows the distribution of allocated tracks from thin devices in the storage group across the thin tiers in the associated FAST policy. By default, only the devices managed by FAST display, either standard or thin devices, based on the policy. Gatekeeper devices are also filtered out.

Managing the FAST controller

The FAST controller is the intelligence of the FAST software. The FAST controller runs its algorithms continuously in the background, checking to see if any FAST policies initiate a performance improvement for the applications managed by those policies.

The FAST controller has two states—Enabled and Disabled. Disabled means that the FAST controller does not do any data movement between tiers based on the FAST policies. Enabled means that the FAST controller performs data movement between tiers based on the policies. You can differentiate between FAST and FAST VP with the enable and disable commands by adding the `-dp` or `-vp` options to the command.

The FAST controller considers EFD as the highest performing tier, FC as the second highest, and SATA the slowest performing tier. External tiers are considered slower than SATA tiers. The FAST controller tries to place the busiest volumes on EFD, as well as make sure it does not exceed the upper limit set by the FAST policy.

In FAST, the FAST controller, an entity residing on the Service Processor, was solely responsible for FAST data movements. The FAST controller gathered and analyzed device statistics then issued explicit movement instructions to the VMAX array.

With FAST VP, there is a division of responsibilities between the FAST controller and the Enginuity operating system. The performance information is gathered by Enginuity and sent to the FAST controller. It issues explicit movement instructions for thin devices, but only for compliance movements. The responsibility for performance movements for thin devices rests with Enginuity. The FAST controller sends to Enginuity a set of partial instructions, called a movement policy, for each storage group under FAST control. Enginuity interprets these instructions and independently executes performance movements for thin devices, based on the device statistics.

Because of these architectural changes, in the event of a loss of the service processor, performance statistics will continue to be collected. Performance data movements will also continue for a short period of time, but will eventually stop, until the service processor is recovered.

This section explains the following FAST controller topics:

"Enabling and disabling FAST" on page 187

"Displaying the FAST state" on page 188

"Managing time windows" on page 189

"Setting FAST control parameters" on page 189

"Displaying FAST control parameters" on page 192

"Displaying FAST plans" on page 193

"Approving a FAST plan" on page 195

"Declining a FAST plan" on page 196

Enabling and disabling FAST

By default, the FAST controller is disabled when shipped. Using the `symfast` command, the FAST controller can be enabled or disabled. To enable the FAST controller, use the `symfast enable` command.

```
symfast -sid SymmID [-i Interval] [-c Count] [-noprompt]

    enable [-dp | -vp]
    disable [-dp | -vp]
```

Where:

`-dp` — Enables/disables FAST disk group provisioning.

`-vp` — Enables/disables FAST VP (virtual pool) provisioning.

If neither option is supplied, both FAST and FAST VP will be enabled/disabled.

Note: Solutions Enabler supports FAST controls for local and remote arrays running Enginuity 5876.

For example, to enable the FAST controller for disk group provisioning, enter:

```
symfast -sid 207 enable -dp
```

To disable the FAST VP controller, enter:

```
symfast -sid 207 disable -vp
```

Note: Depending on the state of the FAST controller, the `disable` action can take some time to complete.

Disabling the FAST controller without the `-dp` or `-vp` options stops all data movement of both standard and thin devices, although if there is currently a FAST standard plan executing, the current group will complete before the FAST controller is disabled. Likewise, if the FAST controller is currently executing thin data movements for compliance, those data movements will complete before the FAST controller is disabled.

If the disable command fails, for example because the FAST controller is unresponsive or DisabledWithError, it is possible that performance movement will continue for thin devices.

In this case, the only way to completely stop the thin data movement is to change the `-thin_data_move_mode` control parameter to NONE. Refer to "Setting FAST control parameters" on page 189.

Displaying the FAST state

To verify the state of the FAST controller, use the `list -state` command, optionally specifying the `-vp` or `-dp` options. If `-vp` or `-dp` are not specified, both options display.

Output displays similar to the following two examples:

```
symfast -sid 234 list -state -dp

Symmetrix ID: 000194900234

    FAST State              : Enabled
    Degraded Reason         : N/A
    Error Message           : N/A
    Current Activities      : Idle

symfast -sid 234 list -state -vp

Symmetrix ID: 000194900234

    FAST State              : Degraded
    Degraded Reason         : No DRVs
                              FAST VP is not licensed
    Error Message           : N/A
    Current Activities      : Idle
```

In the previous output the `FAST State` can have the following possible values:

`Enabled` —The FAST controller is enabled.

`Disabled` — The FAST controller is disabled.

`Disabling` — The FAST controller is in transition from Enabled to Disabled. Or Disable is in progress.

`DisabledWithError` — The FAST controller is disabled with error.

`Degraded` — The FAST controller was activated but not fully functional because of absence of DRV devices or other reasons. It is possible for multiple degraded reasons to occur simultaneously. The `Degraded Reason` field may have a reason for the degraded state or will have a value of `N/A`. The possible values for the `Degraded Reason` are:

- FAST thin tiers reached PRC (Pool Reserved Capacity) value
- FAST VP compliance movement failed
- FAST VP performance movement policy update failed
- FAST algorithm experienced an error
- Optimizer algorithm experienced an error
- Optimizer/FAST standard move time window is not present or does not extend into the future
- FAST thin move time window is not present or does not extend into the future
- Performance time window is not present or does not extend into the future
- Optimizer/FAST movement is not allowed due to internal hard script error
- Optimizer is not licensed
- FAST is not licensed
- FAST VP is not licensed
- Statistics collection is failing for standard device; No Performance movement will happen
- Statistics collection is failing for thin devices; No Performance movement will happen
- No DRVs
- Broken metadevices
- Illegal time windows
- Illegal groups and rules
- Illegal device attributes
- Illegal FAST parameters

The `Error Message` will display more information, if available, about the FAST controller when its state is `DisabledWithError`.

In the previous output the `Current Activities` can have the following possible values:

`Idle` — Fast controller is current idle.

`RunningPlan` — The FAST controller is running a plan.

`FetchingStats` — The FAST controller is fetching statistics.

`AnalyzingStats` — The FAST controller is analyzing statistics.

`PendingPlan` — The FAST controller has a plan scheduled and is waiting to start.

Managing time windows

The `symtw` command allows you to manage time windows. You can use this command to convert any existing time windows to the new format.

Refer to Chapter 18, "Managing Time Windows (Enginuity 5876 or lower)" for information about managing time windows.

Setting FAST control parameters

FAST control parameters are set using the following form:

```
symfast -sid SymmID [-i Interval] [-c Count]
   set -control_parms
         [-approval_mode Auto_Approve | User_Approve] |
         [-min_perf_period PerfTime] |
         [-vp_data_move_mode <Auto | None>]
         [-workload_period WorkTime] |
         [-max_simult_devs MaxSimultDevs] |
         [-max_devs MaxDevs] |
```

```
[-vp_reloc_rate ThinRate]
[-swap_notvisible_devs Enable | Disable]
[-allow_only_swap Enable | Disable ]
[-pool_resv_cap ResvPct]
[-vp_allocation_by_fp <ENABLE | DISABLE>]
[-time_to_compress <NumDays | never>]
[-fast_compression_rate <FastCompRate>]
[-compliance_mode <ENABLE | DISABLE>]
```

Where:

`-approval_mode` — Sets the mode of the FAST controller to automatic or user approval mode using a command line. The possible values are: `Auto_Approve` and `User_Approve`. The default value is `Auto_Approve`. If the FAST controller is set to `User_Approve` mode, it will generate plans, but not perform any movements unless the plans are approved by the user. The `-approval_mode` option can set the mode of the FAST controller and Optimizer using a command line. This parameter is for standard devices only.

Note: The `-approval_mode` option replaces the `-mode` option. The -mode option will continue to work, although its use will be deprecated. The `-mode` option can also be used to set the CLI compatibility mode.

`-min_perf_period` — Specifies the amount of samples required initially before a recommendation is made. You should make sure that the values you specify are long enough (usually a week) for FAST to establish a good characterization of your typical workloads. The parameter exists in case you do not want to wait until the entire workload period has elapsed before the FAST controller commences its analysis and activity. The minimum value is 2 hours, the maximum value is the current value of the workload period parameters.

`-vp_data_move_mode` — The VP data movement mode for thin devices is analogous to the approval mode for standard devices. In Auto mode, the FAST system continuously performs data movement for thin devices within the data movement window, without user intervention. If the mode is set to None, the FAST controller does not perform any data movement for thin devices. There is no equivalent of User_Approve mode for VP data movement.

`-workload_period` — Specifies the amount of workload sampling that FAST should maintain for sample analysis. This is specified in units of time (hours). For example, you can indicate the FAST controller should maintain two weeks worth of workload samples for analysis. The minimum value is 2 hours, the maximum value is 672 (4 weeks).

`-max_simult_devs` — Determines the maximum number of devices that can be moved simultaneously. The minimum value is 2, the maximum value is 32. The default value is 8. This parameter is for standard devices only.

`-max_devs` — Determines the maximum number of devices that can be moved in a 24-hour period. The minimum value is 2, the maximum value is 200. The default value is 200. This parameter is for standard devices only.

`-vp_reloc_rate` — The FAST VP relocation rate determines the aggressiveness of the data movement for thin devices, similar to the `max_simult_devs` setting for standard devices. The lower the value of the VP relocation rate, the more aggressive

FAST will be. This setting does not affect the speed of the data movement itself; that is controlled by the VLUN QoS setting. The minimum value is 1, the maximum value is 10, and the default value is 5.

`-swap_notvisible_devs` — Indicates if the FAST controller can use host invisible devices (unmasked and unmapped) to do a full swap with devices in storage groups under FAST control to improve performance of the storage group. Possible values are ENABLE and DISABLE. The default is DISABLE. This parameter is for standard devices only.

`-allow_only_swap` — Indicates that the FAST controller can only perform a full swap of devices and cannot move devices to unconfigured space. The default is DISABLE. This parameter is for standard devices only.

`-pool_resv_cap` — The pool reserved capacity (PRC) is a percentage of the capacity of each virtual pool that will be reserved for non-FAST activities. If the free space in a given pool (as a percentage of pool-enabled capacity) falls below the PRC, the FAST controller does not move any more chunks into that pool. To move any new chunks to the pool, the FAST controller must first move some chunks from that pool to another pool to free up space. Enforcement of the PRC is best-effort; FAST may move chunks to a virtual pool resulting in a violation of the PRC, because non-FAST activities (such as new allocations for writes to a thin device) can simultaneously consume pool free capacity. The minimum value is 1, the maximum is 80, and the default is 10.

Note: To set the PRC of an individual pool, use the `set pool` command, as explained in "Setting pool attributes (Enginuity 5773 through 5876)" on page 283. If the PRC is not set at the pool-level, the value set with `symfast` is the default.

`-vp_allocation_by_fp` — When set to ENABLE, for all thin devices managed by FAST, Enginuity will choose a pool from the policy when making an allocation for the thin device. When set to DISABLE, the allocation will be from the bound pool. The default value for this option is DISABLE.

Note: The `-vp_allocation_by_fp` option is only supported by Enginuity 5876 and higher. All Enginuity versions prior to 5876 will display N/A for this option.

`-time_to_compress` — Indicates the idle time FAST VP will wait before it compresses devices automatically. The default value is never (not enabled). The enabled value is in days, the range is minimum 40 to 400 days. Any other value returns an error.

Note: If your business works on quarterly basis, it is recommended that you set this value to 100 days to account for end of quarter activities.

`-fast_compression_rate` — Indicates how aggressive the FAST VP controller works when trying to find candidates for compression. The default value for the FAST compression rate is 5. The values accepted for FAST compression rate are 1 (most aggressive) to 10 (least aggressive).

`-compliance_mode` — Sets a higher priority on performing moves required to achieve compliance with the specified FAST policy percentages. This is a global FAST setting which is disabled by default.

For example, to set the FAST controller to `AUTO_APPROVE` mode, the amount of workload sampling for analysis to `200` hours, and the minimum amount of sampling before the recommendation is `100` hours, enter:

```
symfast -sid 207 set -control_parms -mode AUTO_APPROVE -min_perf_period 100
    -workload_period 200
```

For example, to set the FAST controller for the maximum concurrent device movement to `8` and the maximum number of devices that can be moved per day to `32`, enter:

```
symfast -sid 207 set -control_parms -max_simult_devs 8 -max_devs 32
```

Note: Metadevice movement is counted as one device.

Displaying FAST control parameters

To display the FAST control parameters, use the following form:

```
symfast [-sid SymmID] [-offline]
   list -control_parms
```

Note: Before using the `-offline` option, the SYMAPI database must be populated with the storage group, tier, and FAST data using the `symcfg sync -fast` command:

For example, to display the control parameters for VMAX array 432, enter:

```
symfast -sid 432 list -control_parms
```

Output similar to the following displays:

```
Symmetrix ID: 000195700432

    Optimizer and FAST Control Parameters:

    Data Movement Mode                : User_Approve
    Max Simultaneous Device Moves     : 8
    Max Device Moves Per Day          : 200

    Optimizer, FAST and FAST VP Control Parameters:

    Min Initial Workload Period(hrs)  : 2
    Workload Analysis Period(hrs)     : 2

    FAST Control Parameters:

    Swap Not Visible Devices          : Disabled
    Allow Only Swap                   : Disabled

    FAST VP Control Parameters:

    FAST VP Data Movement Mode        : NONE
    FAST VP Data Relocation Rate      : 5
    Thin Pool Reserved Capacity(%)    : 10
    VP Allocation By FAST policy      : Enabled
    FAST VP Time to Compress          : Never
    FAST VP Compression Rate          : 5
```

The features that require Enginuity 5876 Q42012 SR are shown in bold.

These are the possible values for DP data movement mode:

`Auto_Approve` — Indicates the FAST controller is in automatic mode.

`User_Approve` — Indicates the FAST controller is in user approval mode.

These are the possible values for VP data movement mode:

`Auto` — Indicates the FAST controller is in automatic mode.

`None` — Indicates the FAST controller will not perform any data moves on thin devices.

Displaying FAST plans

Note: FAST VP does not display plans for thin devices. However, you can view data movement details in the Optimizer logs (`symoptmz -sid` *`SymmID`* `read -log_type RUNTIME`).

FAST generates optimization plans approximately every hour. FAST plans can contain multiple move/swap suggestions, listed by group. You can query the system at any time to view the current FAST plan.

The FAST controller/Optimizer has the ability to concurrently execute up to two types of plans:

- FAST DP/traditional Optimizer plan (FAST DP plan)
- Optimizer manual swap/rollback plan (Optimizer manual plan)

Plans are subdivided into groups that execute serially. Each individual plan has a unique ID.

The `symfast` command does not manage Optimizer manual plans, except to show the history of these activities. Use the following form to display a FAST plan:

```
symfast -sid SymmID
   list -plan [-v]
```

Note: Both Optimizer plans and FAST plans will display with the `symfast list -plan` command. Check the `Group Attributes` in the output for the origin of the plan.

For example, to list the FAST plan for VMAX array `234`, enter:

```
symfast list -sid 234 -plan
```

Output similar to the following displays:

```
Symmetrix ID                    : 000194900234

Plan ID                         : 12222011:154359
Plan Type                       : Auto Generated
Plan State                      : CnfigInProgress
Start Time                      : Tue Dec 22 20:30:23 2011
Percent Complete                : 5%
Estimated time to completion    : 04:12:30
Number of Groups                : 3
  . . .
Group 2:
  {
  Group Attributes              : FAST Generated(Performance)
  Group State                   : NotStarted
  Time Started                  : N/A
  Time Completed                : N/A
```

```
  Percent Complete             : 0%
  Estimated time to completion : 01:35:31

  Swap Pairs (2)
    {
              Source Device                          Target Device
    ------------------------------------ ------------------------------------
                              Storage                                Storage
    Sym   Tier Name  Prot     Group Name  Sym   Tier Name  Prot     Group Name
    ----- ---------- -------- ----------- ----- ---------- -------- -----------
    00045 PrimeTier  R1       OraSales    000E0 PrimeDBTi* R5(3+1)  OraSales
    00046 PrimeTier  R1       OraSales    000E1 PrimeDBTi* R5(3+1)  OraSales
    }
  }

Group 3:
  {
  Group Attributes             : FAST Generated(Compliance)
  Group State                  : NotStarted
  Time Started                 : N/A
  Time Completed               : N/A
  Percent Complete             : 0%
  Estimated time to completion : 04:12:30

  Move Devices (6)
    {
                    Source Device                            Target
    ------------------------------------------------ --------------------------
                                Dsk Storage                                  Dsk
    Sym   Tier Name     Prot     Grp Group Name      Tier Name     Prot     Grp
    ----- ------------- -------- --- --------------- ------------- -------- ---
    00042 N/A           R6(14+2) 003 OraSales        ArchiveDBTier R5(7+1)  004
    00043 N/A           R6(14+2) 003 OraSales        ArchiveDBTier R5(7+1)  004
    00044 N/A           R6(14+2) 003 OraSales        ArchiveDBTier R5(7+1)  004
    00059 PrimeDBTier   R5(3+1)  002 OraSales        WorkDBTier    R5(3+1)  003
    0005A PrimeDBTier   R5(3+1)  002 OraSales        WorkDBTier    R5(3+1)  003
    0005B PrimeDBTier   R5(3+1)  002 OraSales        WorkDBTier    R5(3+1)  003
    }
  }
```

FAST plan output description

In the previous display, the `Group Attributes` can have the following values:

`FAST Generated` — Indicates the swap/move plan was generated by the FAST controller.

`Optimizer Generated` — Indicates the swap/move plan was generated by the Symmetrix Optimizer.

In the previous display, the `Group State` can have the following values:

`Done` — The group has finished everything.

`InProgress` — The group is currently running. If running, the `Percent Complete` and the `Estimated time to completion` fields report the progress.

`NotStarted` — The group has not started running yet.

`Failed` — The execution of the entire group has failed and will be retried.

In the previous display, the `Plan State` field can have the following possible values in `USER_APPROVE` mode and will be set to `NA` in `AUTO_APPROVE` mode:

`NotApproved` — Indicates that this is a proposed plan for the FAST controller and the user needs to approve it before the FAST controller can implement the plan.

`ApprovedWithSpecifiedTime` — Indicates that the plan has been approved and scheduled at a specific time.

`ApprovedWithDelay` — Indicates that the plan has been approved and is waiting in a queue to run.

`ApprovedWithConfigParameters` — Indicates that the plan has been approved and will be scheduled as per the time window and control parameter settings.

`InProgress` — Indicates that the plan is currently running.

`Aborting` — Indicates that the plan is currently aborting. This state can occur when the user attempts to decline a running plan.

For FAST DP plans, the plan listing also displays source tier name, source disk group number, and source RAID protection type for device moves. The target disk group name longer displays for device moves, although target disk group number still displays. Time started and completed also displays for each group.

Approving a FAST plan

When FAST is in `USER_APPROVE` mode, it generates plans approximately every hour, but does not act on them. A FAST plan can be approved or declined. FAST only executes approved plans. Query the system often to be shown the current FAST plan.

Note: FAST plans cannot be partially approved. Plans are approved or declined as a whole.

Use the `symfast -plan approve -id PlanID` command to approve a FAST plan. If you want the plan to start at a specific time, you can add the `-begin_at TimeVal` option to the command.

```
symfast -plan -sid SymmID [-i Interval] [-c Count] [-noprompt]
   approve -id PlanID [-begin_at=TimeVal]
```

To identify the *PlanID*, use the `Plan ID` returned in the `symfast list -plan` output. The `-begin_at` option specifies the specific time to run the FAST plan. If this option is not specified, FAST defaults to the time window definition. The *TimeVal* format is `MMDDYYYY:HHMMSS`.

Note: Use the `-i` and `-c` options to monitor FAST progress over time.

For example, to approve the FAST plan `08022009:150115` on VMAX array `207`, and execute this plan during the scheduled time window, enter:

```
symfast -plan -sid 207 approve -id 08022009:150115
```

The FAST plan 08022009:150115 is approved.

Note: The `approve` command can only be used in `USER_APPROVE` mode.

Declining a FAST plan

To decline a FAST plan, use the following form:

```
symfast -plan -sid SymmID [-i Interval] [-c Count] [-noprompt]
   decline -id PlanID
```

A running plan can be declined. The FAST controller will attempt to abort a running plan; it will complete the current group, and not start any other group of changes.

For example, to decline the FAST plan `08022009:150115` on VMAX array `207`, enter:

```
symfast -plan -sid 207 decline -id 08022009:150115
```

The FAST controller deletes the plan. A plan can be declined both before approval and after approval. An approved plan can be declined while in progress, but the current in progress group will run to completion.

The `decline` action of `symfast` allows an in-progress FAST standard plan to be aborted, even in AUTO_APPROVE mode. Aborting a plan deletes all groups except for the in-progress group. The FAST controller can immediately generate and begin execution of a new plan unless it is disabled or the approval mode is changed to USER_APPROVE.

Note: The `decline` command can be used in either `USER_APPROVE` or `AUTO_APPROVE` mode to abort a running plan.

Displaying FAST information

This section contains list and show displays to use to examine FAST data. Reports about the following topics are explained:

"FAST activity history" on page 196

"Audit log messages" on page 197

The next section, "FAST reports" on page 197, provides additional examples of data that can be generated about FAST configurations.

FAST activity history

Note: FAST VP does not show thin device history.

Use the following syntax to display the activity history, or the FAST plan for a VMAX array:

```
symfast [-sid SymmID] list -history [-v]
   [-start_date TimeVal] [-end_date TimeVal]
```

Optionally use the `-start_date` and `-end_date` to narrow down the time range.

For example, to view the FAST activity history for VMAX array 234, enter:

```
symfast -sid 234 list -history
```

Output similar to the following displays:

```
Symmetrix ID                : 000194900234
Number of Groups            : 4
   . . .
Group 3:
```

```
  {
  Time Started             : Tue Dec 22 20:40:50 2011
  Time Completed           : Tue Dec 22 20:45:23 2011
  Attributes               : FAST Generated(Performance)

  Swap Pairs (2)
    {
              Source Device                          Target Device
    ------------------------------------ ------------------------------------
                              Storage                                Storage
    Sym   Tier Name  Prot     Group Name  Sym   Tier Name  Prot     Group Name
    ----- ---------- -------- ----------- ----- ---------- -------- -----------
    00045 PrimeTier  R1       OraSales    000E0 WorkDBTier R5(3+1)  OraSales
    00046 PrimeTier  R1       OraSales    000E1 WorkDBTier R5(3+1)  OraSales
    }
  }

Group 4:
  {
  Time Started             : Tue Dec 22 20:50:23 2011
  Time Completed           : Tue Dec 22 21:05:20 2011
  Attributes               : FAST Generated(Compliance)

  Move Devices(6)
    {
                    Source Device                            Target
    ------------------------------------------------ --------------------------
                                Dsk Storage                                  Dsk
    Sym   Tier Name     Prot     Grp Group Name      Tier Name     Prot     Grp
    ----- ------------- -------- --- --------------- ------------- -------- ---
    00042 N/A           R6(14+2) 003 OraSales        ArchiveDBTier R5(7+1)  004
    00043 N/A           R6(14+2) 003 OraSales        ArchiveDBTier R5(7+1)  004
    00044 N/A           R6(14+2) 003 OraSales        ArchiveDBTier R5(7+1)  004
    00059 PrimeDBTier   R5(3+1)  002 OraSales        WorkDBTier    R5(3+1)  003
    0005A PrimeDBTier   R5(3+1)  002 OraSales        WorkDBTier    R5(3+1)  003
    0005B PrimeDBTier   R5(3+1)  002 OraSales        WorkDBTier    R5(3+1)  003
    }
  }
```

Audit log messages

The following FAST user actions will have entries in the VMAX array audit log:

- Create/delete a tier; add/remove disk groups
- Create/delete a policy; add/remove tiers; rename a policy; change a policy tier capacity
- Associate/disassociate a storage group to/from a policy; change the priority of a storage group association
- Enable/disable the FAST controller; approve/decline a plan
- Set FAST control parameters

FAST reports

FAST provides two types of reports about the FAST configuration. The compliance report and technology demand report together show whether the most efficient use of storage is in place.

Compliance report — Displays compliance information for all storage groups associated with FAST policies, including policies containing VP tiers and policies containing DP tiers. The compliance report examines compliance for each FAST storage group in isolation; contention for tier capacity among storage groups is not accounted for, nor is tier capacity consumed by devices not under FAST control. See "Compliance report" on page 198.

Technology demand report — Provides information about this contention for tier resources. See "Technology demand reports" on page 200.

Compliance report

A storage group is FAST-compliant only if all devices in the storage group exist only on the tiers defined in the policy and the percentage capacity of all the tiers occupied by the storage group are within the upper limits of the tier capacities specified in the policy.

The `symfast` command provides options for generating reports about storage groups, policies, and the current logical capacity demand of that association. To generate a report for the demand of a storage group and its associated FAST policy, use the following form:

```
symfast -sid SymmID [-offline]
   list -association [-demand [-sg SgName |
                      -fp_name PolicyName][-mb]]
```

Where:

`-association` — Displays FAST associations that exist between storage groups and FAST policies.

`-demand` — Displays details about the demand of FAST storage groups on storage tiers.

`-sg` — Displays a report for a specific storage group.

`-fp_name` — Displays a report for a specific FAST policy.

`-mb` — Displays the reported capacities in MBs instead of the default (GB).

Note: Prior to using the `-offline` option, the SYMAPI database must be populated with tier, storage group, and FAST data using the command `symcfg sync -fast`.

For storage groups associated with policies containing DP tiers, the limit, `FAST SG Used`, and `Growth` columns will be based on the logical capacity of standard devices in the storage group only. The capacity of any thin devices in the storage group will not be counted.

For storage groups associated with policies containing VP tiers, the limit, `FAST SG Used`, and `Growth` columns will be based on the allocated capacity of thin devices in the storage group only. The capacity of any standard devices in the storage group will not be counted, nor will any unallocated logical capacity of the thin devices. The display also indicates whether the RDF Coordination value is set for each association. For FAST disk group associations, this value is always `N/A`. For FAST VP associations, `Enable` or `Disable` displays.

The following example provides output for the `list -association -demand` command:

```
symfast list -sid 432 -association -demand

Symmetrix Id     : 000194900432

Policy Name      : HR_DP
Storage Group    : HR_DP
Priority         : 1
RDF Coordination : N/A

Tiers (1)
   {
   ---------------------------------------------------------------------
                                          Logical Capacities (GB)
                                       -------------------------------
                       Target    Max SG     Max SG   FAST SG
   Name          Type Prot     Percent     Demand     Usage     Growth
   ------------- ---- -------- ------- ---------- --------- ----------
   HR_EAV        DP   R1           100        111         0       +111
   [OutOfPolicy] N/A  N/A            -          -       111       -111

       Total                           ---------- ---------
                                              111       111
   }
Policy Name      : HR_FP
Storage Group    : HR_SG
Priority         : 3
RDF Coordination : Enabled

Tiers (1)
   {
   ---------------------------------------------------------------------
                                          Logical Capacities (GB)
                                       -------------------------------
                       Target    Max SG     Max SG   FAST SG
   Name          Type Prot     Percent     Demand     Usage     Growth
   ------------- ---- -------- ------- ---------- --------- ----------
   HR_TIER       VP   R1           100          4         2         +2

       Total                           ---------- ---------
                                                4         2
   }

Legend:
  Tier Type     :  DP = Disk Group Provisioning, VP = Virtual Pools
```

Compliance report column descriptions

The compliance report columns provide the following information:

[`OutofPolicy`] row in the `Name` column — Shows the devices in the storage group that currently do not reside on the tiers defined in the FAST policy.

`Max SG Percent` column — Shows the limit of the storage group per tier as defined in the FAST policy.

`Max SG Demand` column — Shows the calculated upper limit, in GB, for the storage group on the tier.

`FAST SG Usage` column — Shows the current occupancy of the storage group in a tier.

`Growth` column — Shows, as per the FAST policy, how much more the storage group can grow on a given tier. This column also indicates compliance. If the potential to grow is negative, the storage group has exceeded the capacity limit for this tier, therefore the storage group is out of compliance.

Technology demand reports

Technology demand reports show the state of the FAST configuration from the point of view of technology and tiers. The thin demand report presents a VMAX array-wide snapshot of the current allocations of thin devices under FAST control and of the capacities of thin pools in thin tiers. The report shows the demand the thin devices place on each of the pools if no new allocations were made.

To generate a report for the demand (or details about) a technology type and its tiers, use the following form:

```
symfast [-sid SymmID] [-offline]

   list <-tech <EFD | FC | SATA | ALL> -external>
   -demand [-v] [-dp | -vp][-allocated]
```

By default, the maximum storage group demand is calculated using the logical (configured) capacity of thin devices under FAST control. If `-allocated` is specified, Max SG Demand is calculated using the allocated capacity of thin devices under FAST control.

When specifying `-external`, do not use with the `-technology` option.

If neither `-dp` or `-vp` is specified, `-dp` is the default. If the array only contains VP tiers, and the `-vp` option is not specified, the report will show the VP tiers.

Note: Prior to using the `-offline` option, the SYMAPI database must be populated with tier, storage group, and FAST data using the command `symcfg sync -fast`.

The following example shows the command and output for a SATA tier:

```
symfast list -sid 432 -tech SATA -demand -vp

Symmetrix ID : 000195700432

Technology   : SATA

VP Tiers (4)
   {
   ------------------------------------------------------------------------------
                  A                          Logical Capacities (GB)
                  T           ---------------------------------------------------
                  T Target       Tier    Tier    Tier FAST SG   FAST Max SG Excess
   Tier           R Prot      Enabled    Free    Used   Usage  Avail Demand
   ------------- - -------- ------- ------- ------- ------- ------ ------ ------
   HR_test        N R1           889     883       6       0      0      -      -
   PAYRL_test_t*  N R1             4       4       0       0      0      -      -
   PAYRL_tier     F R1             4       4       0       0      3      8     -5
   RECV_366859    N R1             0       0       0       0      0      -      -
    Total                     ------- ------- ------- ------- ------ ------ ------
                                   897     891       6       0      3      8     -5
   }

Legend:
  ATTR       :  F = Tier in a FAST policy associated with SG(s)
             :  P = Tier in a FAST policy unassociated with SG(s)
             :  N = Tier not in any FAST policy
```

The VP demand report displays information only for VP tiers and for storage groups associated with FAST policies containing VP tiers. The `symfast list -external -demand -vp` display adds another block for external tiers. The technology will display `N/A` for external tiers.

```
symfast -sid 432 list -external -demand -vp

Symmetrix ID : 000194900432

Technology   : N/A
Disk Location: External

VP Tiers (1)
   {
   ---------------------------------------------------------------------------
                 A                        Logical Capacities (GB)
                 T          ---------------------------------------------------
                 T Target      Tier    Tier    Tier FAST SG   FAST Max SG Excess
   Tier          R Prot     Enabled    Free    Used   Usage  Avail Demand
   ------------- - -------- ------- ------- ------- ------- ------ ------ ------
   external_test N Unprot       894     888       6       0      0      -      -
    Total                    ------- ------- ------- ------- ------ ------ ------
                                894     888       0       0      0      -      -
   }

Legend:
  ATTR      :  F = Tier in a FAST policy associated with SG(s)
            :  P = Tier in a FAST policy unassociated with SG(s)
            :  N = Tier not in any FAST policy
```

The columns in the previous output example are defined as follows:

`Tier` — Shows names of VP tiers.

`ATTR` — Shows the status of the VP tier. Tiers can have one of 3 possible attributes:

- In a FAST Policy associated with a storage group (`F`)
- In a FAST Policy or Policies where none of the FAST Polices are associated with a storage group (`P`)
- Not in any FAST Policy (`N`)

`Target Prot` — Target protection of the tier.

`Tier Enabled` — Total logical capacity enabled for the tier.

`Tier Free` — The enabled logical capacity minus the used logical capacity, minimum 0.

`Logical Tier Used` — Total pool allocated capacity for all thin pools in the tier. Allocated capacity on all DATA devices will be counted, including DATA devices that are not enabled; therefore Used may be greater than Enabled.

`Logical FAST SG Usage` — Sum of allocated capacity residing on this tier from thin devices in a storage group associated with a FAST Policy containing VP tiers. A separate [OutOfTier] line lists the allocated capacity from such thin devices in a pool of matching technology type that is not included in any VP Tier.

`Logical FAST Available` — If the thin tier is in a FAST policy associated with a storage group, the FAST Available capacity is equivalent to the Tier Free value less the PRC from all thin pools included in the thin tier (with a minimum value of zero) plus the FAST SG Usage. If the tier is not in any FAST policy or is in policies where none of the policies are associated to a storage group then this value will be 0.

`Logical Max SG Demand` — Sum of the allocated capacity of all thin devices in a storage group associated with a FAST policy containing this VP tier if the devices were to occupy the full allotted quota (per the limit defined in the FAST policy) of space in the VP tier. Logical Max SG Demand also equals them sum of the values in the Max SG

Demand column for all entries for this tier in the compliance report. If the tier is not in any FAST policy or is in policies where none of the policies are associated to a storage group then this value is not applicable.

`Logical Excess` — Difference between FAST Available and Logical Max SG Demand. If the thin tier is not in any FAST policy or is in policies where none of the policies are associated to a storage group then this value is not applicable.

CHAPTER 7
Managing Configuration Changes

This chapter describes configuration change concepts and explains how to perform change operations using the `symconfigure` command.

- Introduction 204
- Verifying and checking sessions 206
- Making configuration changes safely 207
- Supported configuration operations 209
- Setting array attributes 213
- Managing external disk groups 214
- Creating devices (HYPERMAX OS 5977) 214
- Creating devices (Enginuity 5773 through 5876) 217
- Copying devices 226
- Managing metadevices (Enginuity 5773 through 5876) 228
- Configuring thin metadevices (Enginuity 5773 through 5876) 235
- Converting devices 236
- Setting device attributes 241
- Setting device geometry (Enginuity 5773 through 5876) 243
- Setting device identifiers 245
- Deleting devices 247
- Managing device reservations 248
- Managing device pools (HYPERMAX OS 5977) 251
- Managing device pools (Enginuity 5773 through 5876) 251
- Mapping devices 256
- Mapping CKD devices (Enginuity 5773 through 5876) 257
- Unmapping devices 259
- Introducing devices to a host 261
- Configuring dynamic RDF-capable devices (Enginuity 5773) 262
- Setting SRDF group attributes 263
- Swapping RA groups 263
- Converting directors (Enginuity 5876 only) 264
- Port to director emulation support (HYPERMAX OS 5977) 265
- Setting port characteristics 268

Introduction

This chapter explains how to perform change operations using the SYMCLI `symconfigure` command. You can use `symconfigure` to perform control operations on arrays, as well as the array devices, groups, directors, and ports. Array controls include setting array-wide metrics and determining what type of devices the array will support, such as RAID 6 devices. Device controls include creating devices, mapping and masking devices, and configuring device pools. You can also use `symconfigure` to reserve devices and release device reservations.

You can invoke `symconfigure` from the local host to make configuration changes to a locally-connected array, or to an RDF-linked array.

Configuration change session rules

- Concurrent Provisioning — Up to four concurrent configuration change sessions are allowed to run at the same time, when they are non-conflicting. This means that multiple parallel configuration change sessions can run at the same time as long as the changes do not include any conflicts on the following:
  - Device back-end port
  - Device front-end port
  - Device
- The array manages its own device locking.
- A session ID identifies each running session on the array.

For arrays running Enginuity 5773 through 5876, devices and metadevices can be created and mapped in one command. This eliminates the need to perform multiple configuration changes when provisioning new storage.

Symconfigure command execution formats and options

The `symconfigure` command can be executed in the following formats:

- Placing commands, terminated with a semicolon (;), in a command file (most commonly used). For example:

  ```
  symconfigure commit -sid 3160 -file unmap_dev.cmd
  ```

  Where `unmap_dev.cmd` contains:

  ```
  unmap dev 00020:00024 from dir ALL:ALL;
  ```

  Note: When using the `symconfigure -file` option, text files can have a maximum comment of 512 characters on Windows. Make sure the comment line does not exceed 512 characters.

- Redirect a number of screen entries to `stdin` to save keystroke entries (for UNIX platforms), and not use a command file.

  For example, to prepare a chain of `symconfigure` commands on the screen to be redirected to `stdin`, use the following form:

```
symconfigure -sid SymmID prepare <<DELIM
   create dev count=3 size=3200 cyl,
    emulation=FBA, config=2-Way-Mir;
   create dev count=1, size = 3200 cyl,
    emulation=FBA, config=unprotected;
    DELIM
```

- Use the `-cmd` option. With this option, the commands that would normally be put in a command file are enclosed in quotes. A command can run over many lines, but you cannot press **Enter**. For example:

```
symconfigure -sid 256 -cmd "create dev count=3, size = 3200 cyl,
emulation=FBA, config=2-Way-Mir;create dev count=1, size = 3200 cyl,
emulation=FBA, config=unprotected;" -v -nop preview
```

Configuration changes that are submitted to the array are processed in a session. The `symconfigure` command file format contains various command entries terminated with a semicolon (;). You can make multiple changes in one session, but all changes must fall into one complete operation — for example, creating a device, adding the device to a device pool, and enabling the device state. Additional command file examples are provided with the actions described later in this chapter.

Table 6 describes the `symconfigure` command options:

Table 6 Symconfigure command options

symconfigure option	Definition
`abort`	Gains control of an existing session to abort it and free any held locks.
`commit`	Attempts to apply the changes defined in the command file into the specified array.
`list`	Lists the relevant details, depending on the option: • `-freespace` shows the free physical disk space within the array as it can be used to create new devices for different emulation modes. Free disk space on unformatted disks is shown as available for all emulation modes. If a physical disk has been partially used to create a device, that device is considered formatted and the rest of the available space can only be used for devices of the same emulation mode. • `-v` displays configuration information that is not stored in the SYMAPI database and that needs to be retrieved directly from the configuration server. • `-reserve` shows a summary of all reservations.
`prepare`	Validates the syntax and correctness of the operations. Verifies the validity of the command file changes and their appropriateness for the specified array. The prepare action has no function for pool sessions.
`preview`	Ensures the command file syntax is correct. Verifies the validity of the command file changes.
`query`	Returns information about the status of a configuration change session.
`release`	Releases the specified device reservation.
`reserve`	Processes the command file to reserve the indicated devices and displays the resulting reserve ID.
`show`	Shows the details of the specified device reservation.
`verify`	Verifies that the configuration currently running in the specified array complies with the requirements for host-based configuration changes.

Refer to the *EMC Solutions Enabler SYMCLI Command Reference* for the complete manpage description of the `symconfigure` command.

Verifying and checking sessions

You can verify and check a session acting on a specified `symconfigure` command file using the `verify`, `preview`, and `prepare` arguments. The `commit` argument performs these same checks, but then attempts to execute the specified action.

The `symconfigure` command allows you to modify aspects of a Symmetrix configuration. A lock may be taken out by the specified Symmetrix configuration server while the configuration change session is active. A session is processed in a series of stages which may be monitored using the query option.

Not all stages are always executed. Use discretion when controlling which stages are to be completed to allow checking and debugging of the command file before the changes are implemented.

Verify

There are restrictions as to what state the Symmetrix configuration must be in before allowing changes to be applied from a host system. To determine if a Symmetrix array can be modified, use the `verify` command.

Preview

The `preview` option verifies the syntax and correctness of each individual change defined, their correctness as a set, and then terminates the session without change execution (correct within the realm of the host and valid as a possible array configuration).

Prepare

The `prepare` option performs the preview checks, validates the change operation (devices are in correct state, etc.), and also verifies the appropriateness of the resulting configuration definition against the current state of the arrays. The argument then terminates the session without attempting to make the configuration change.

Note: For arrays running Enginuity 5773 through 5876, the `prepare` argument is not supported for SAVE device pool sessions.

Commit

The `commit` option completes all checks and verifications, and then attempts to make the requested configuration changes in the specified array.

Query

The `query` argument allows you to check the status of any configuration change session. After you commit a command file for execution, it takes a few minutes to complete execution.

For example, to check the status of the change session for the next 2 minutes, every 10 seconds on the array `345`, enter:

```
symconfigure -sid 345 query -i 10 -c 12
```

The `query` option is useful in SRDF environments, where a change to a local array on one host results in a corresponding change to a remote array. The System Manager of a host connected to the remote array can monitor the progress of the change. A `query` is also helpful at sites where the Symmetrix Optimizer is also modifying a configuration by rearranging the placement of data.

You can also use `query` to show if there is a configuration session running.

Abort

The `abort` option allows you to abort a configuration session. To abort a change session on array `12345`, enter:

```
symconfigure -sid 12345 abort
```

Because changes made in the SRDF operations class will initiate actions on the local and remote arrays, it might become necessary to abort processing on a remote array.

At some point during `commit` processing, a *point of no return* is reached. Any attempt to abort will be denied once processing has reached this point.

You can add the `-session_id` option to the `symconfigure abort` command to differentiate between multiple running sessions. To abort the change session `100` on array `343`, enter:

```
symconfigure -sid 343 abort -session_id 100
```

If the `session_id` is not known, running the `symconfigure -sid SymmID abort` command will display each running session and prompt for the session ID.

Making configuration changes safely

Before making configuration changes, you must thoroughly understand your array configuration. The following guidelines help to establish safe disciplines as the changes you make can easily impact stored data:

- Verify that the current array configuration is a viable configuration for host-initiated configuration changes.

  To verify your current array configuration is ready for changes, enter:

  ```
  symconfigure verify -sid SymmID
  ```

- Before creating new devices, check for free physical disk space using the following command:

  ```
  symconfigure list -freespace [-units CYL | MB] -sid SymmID
  ```

  Free disk space on unformatted disks is shown as available for all emulation modes. New devices will be created first on physical disks that have no prior allocations, causing these disks to be committed to that emulation type.

  Examine the distribution of free space across formatted disks to see if the desired mirroring can be provided using one of the following commands:

  ```
  symdev list -sid SymmID -da all -space
  or
  symdisk list -sid SymmID
  ```

- Configuration changes only begin after you `commit` the action.

  Some classes of change operations may or may not impact current I/O. When possible, before you `commit` any action, stop I/O activity on the affected devices.

  If required, set the impacted devices for change to be Not Ready using the following command:

  ```
  symdg -g DgName not_ready [LdevName [LdevName...   ]]
  ```

  Note: If I/O activity on an affected device occurs before or during a `commit` action, the `commit` action might fail. At the very least, heavy I/O activity on unaffected devices will impact how long it takes to commit changes.

- Ensure that all your critical data is preserved and safe when creating new or changing device configurations. Do not store data on any device that is not mirrored, or RAID-protected.

  All final configurations and device attribute adjustments must meet certain open systems guidelines detailed in the E-Lab™ Interoperability Navigator, which can be viewed at http://elabnavigator.EMC.com.

  Contact the EMC® Customer Support Center for assistance in reverting to your previous configuration should there be unforeseen problems with the new configuration.

- After committing a `symconfigure` mapping operation, you must update the device mapping information within the host system environment. Attempting host activity with a device after it has been removed or altered, but before you have updated the host's device information, can cause host errors.

  To update your hosts, run the utilities specified for your platform as described in"Introducing devices to a host" on page 261. The `symcfg discover` command should then be issued to update the SYMAPI database with the new device mapping information. After the host environment is updated, I/O activity can resume with the device.

Mixed array environments

Before issuing a `symconfigure` command to an array containing both open system and mainframe devices, you must verify that the mainframe Missing Interrupt Handler (MIH) period for *each* mainframe server attached to the array is set to at least 2 minutes.

To view the current MIH period, use the z/OS command `D IOS,MIH`, and note the value for the DASD device class (for example, `DASD=1:30`).

To change the MIH period, use the z/OS command `SETIOS MIH,DASD=mm:ss`, where *`mm:ss`* is a period of time in minutes and seconds. Then use the z/OS command `D IOS,MIH` to verify your changes.

Once you have completed your configuration change session, use the `SETIOS MIH,DASD=mm:ss` command to set the MIH period back to its original value.

For more information, refer to your mainframe system administrator.

Check Change Tracker session priority (Enginuity 5876 or lower)

The following `symconfigure` operations will be blocked when there is an active Change Tracker session on the related devices:

- Converting to VCMDB device
- Forming and dissolving a metadevice
- Adding metamember to a metadevice
- Removing metamembers from a metadevice
- Converting a metadevice from concatenated to striped
- Deleting a device
- Unbinding a thin device

Supported configuration operations

The supported microcode levels for the configuration change operations are listed in Table 7.

Table 7 Configuration change operations (page 1 of 4)

Configuration change operation	5773	5874	5875	5876	HYPERMAX OS 5977
Create device:					
FBA	✓	✓	✓	✓	✓
Celerra® FBA	✓	✓	✓	✓	✓
RAID-5	✓	✓	✓	✓	
RAID-6	✓	✓	✓	✓	
CKD	✓	✓	✓	✓	
CKD meta devices	✓	✓	✓	✓	
AS/400	✓	✓	✓	✓	AS400_D910_099
VME 512 FBA	✓	✓	✓	✓	
VDEV	✓	✓	✓	✓	
SAVE devices (for VDEVs)	✓	✓	✓	✓	
Thin devices	✓	✓	✓	✓	✓
DATA devices	✓	✓	✓	✓	
DLDEV devices		✓	✓	✓	
Expand device					
Copy device					✓
Add gatekeeper				✓	✓
Delete device	✓	✓	✓	✓	✓

Table 7 Configuration change operations (page 2 of 4)

Configuration change operation	5773	5874	5875	5876	HYPERMAX OS 5977
Set device attributes:					
VCMdb	✓				
ACLX		✓	✓	✓	
RDB_Cksum	✓				
dyn_rdf	✓	✓	✓	✓	
dyn_rdf1_only	✓	✓	✓	✓	
dyn_rdf2_only	✓	✓	✓	✓	
SCSI3_persist_reserv	✓	✓	✓	✓	✓
RCVRPNT_TAG			✓	✓	
DIF1				✓	✓
AS400_GK				✓	✓
Set device geometry	✓	✓	✓	✓	
Set device identifiers	✓	✓	✓	✓	✓
Convert device:					
Adding or removing BCV	✓	✓	✓	✓	✓
Adding or removing RDF	✓	✓	✓	✓	✓
Dynamic RDF	✓	✓	✓	✓	
Converting static SRDF to dynamic SRDF	✓	✓	✓	✓	
Adding another mirror	✓				
Removing a mirror	✓				
Change device emulation:					
FBA and Celerra FBA	✓	✓	✓	✓	✓
Convert director type:					
FA to RDF dir/RDF to FA				✓	
Virtual Provisioning:					
Create thin devices	✓	✓	✓	✓	✓
Create thin pools	✓	✓	✓	✓	
Set pool limits	✓	✓	✓	✓	
Bind thin devices to pools	✓	✓	✓	✓	
Unbind thin device from pool	✓	✓	✓	✓	
Rebind thin device to pool			✓	✓	

Table 7 Configuration change operations (page 3 of 4)

Configuration change operation	5773	5874	5875	5876	HYPERMAX OS 5977
Start/stop free thin device (unwritten allocated space)		✓	✓	✓	
Start/stop free thin device (zero-based reclaim)		✓	✓	✓	
Start/stop drain thin pool		✓	✓	✓	
Rebalance thin pool		✓	✓	✓	
Start/stop allocate tdev		✓	✓	✓	
Rename thin pool			✓	✓	✓
Maximum subscription percent		✓	✓	✓	
Pool reserve capacity		✓	✓	✓	
Thin pool compression				✓	
Automated pool rebalancing		✓	✓	✓	
View thin pools		✓	✓	✓	✓
Map device:					only on non-ACLX enabled ports
AS/400	✓	✓	✓	✓	✓
CKD	✓	✓	✓	✓	
FBA	✓	✓	✓	✓	✓
Manage PAV Alias	✓	✓	✓	✓	
Unmap device:					
AS/400	✓	✓	✓	✓	✓
CKD	✓	✓	✓	✓	
FBA	✓	✓	✓	✓	✓
Form meta	✓	✓	✓	✓	
Add metamember:					
Concatenated	✓	✓	✓	✓	
Striped	✓	✓	✓	✓	
Remove metamember:					
Concatenated	✓	✓	✓	✓	
Dissolve Meta	✓	✓	✓	✓	
Swap RA group	✓	✓	✓	✓	
Swap hyper (Optimizer)	✓				
Swap SRDF device:	✓	✓	✓	✓	
Dynamic SRDF devices	✓	✓	✓	✓	

Table 7 Configuration change operations (page 4 of 4)

Configuration change operation	5773	5874	5875	5876	HYPERMAX OS 5977
Set front-end port attributes	✓	✓	✓	✓	✓
Associate/disassociate a port to director emulation					✓
Set array-wide parameters:					
auto_meta	✓	✓	✓	✓	
auto_meta_config	✓	✓	✓	✓	
auto_meta_member_size	✓	✓	✓	✓	
min_auto_meta_size	✓	✓	✓	✓	
max_hypers_per_disk	✓				
fba_multi_access_cache	✓				
dynamic_rdf	✓				
pav_alias_limit	✓	✓	✓	✓	
PAV_mode	✓	✓	✓	✓	
rdfa_cache_percent	✓	✓	✓	✓	✓
rdfa_host_throttle_time	✓	✓	✓	✓	✓
hot_swap_policy	✓				
Replace disk				✓	
SRDF operations:					
Add/remove SRDF mirror	✓	✓	✓	✓	
Add RDF mirror					✓
Remove RDF mirror					✓
Device pools:					
Create/delete pools	✓	✓	✓	✓	
Add/remove/enable/disable pool members	✓	✓	✓	✓	
Rename a pool			✓	✓	✓
Federated Tiered Storage (FTS):					
Add external disk				✓	✓
Add external disk group				✓	✓
Remove external disk group				✓	✓

Setting array attributes

You can set certain attributes to control array behavior using the following command:

```
set Symmetrix MetricName=MetricValue
            [, MetricName=MetricValue];
```

Where:

MetricName — The metric to be set. Possible metric names are listed in Table 8 and Table 9.

Table 8 Possible metric names (HYPERMAX OS 5977)

rdfa_cache_percent	Sets the percentage of write pending cache that can be used by SRDF/A. This is a value from 0 to 100 percent.
rdfa_host_throttle_time	Sets the number of seconds to throttle host writes to SRDF/A devices when cache is full, before dropping RDF/A sessions. Throttling delays a write from the host until a cache slot becomes free. Values are from 0 to 65535.
pav_alias_limit	If PAV is enabled, specifies the maximum number of aliases that can be assigned to a device. Possible values are:
PAV_mode	Enables the use of PAV (Parallel Access Volumes). Possible values are: STANDARD: Standard PAV volumes with static aliasing can be configured. DYNAMIC_STANDARD: Standard PAV volumes with dynamic aliasing can be configured.

Table 9 Possible metric names (Enginuity 5773 through 5876)

auto_meta	Enables the auto meta feature. This can be enabled only if the other auto meta parameters min_auto_meta_size, auto_meta_config and auto_meta_member_size are set to valid values. Possible values are ENABLE or DISABLE. This option is not applicable for CKD metadevices.
auto_meta_config	Specifies the default meta config when the auto_meta feature is enabled. Possible values are CONCATENATED, STRIPED, or NONE.
auto_meta_member_size	Specifies the default meta member size in cylinders when the auto_meta feature is enabled. Possible values are listed in Table 13 on page 229.
fba_multi_access_cache	Determines whether a read request can share cache slots in some conditions. Possible values are ENABLE or DISABLE.
min_auto_meta_size	Specifies the size threshold that triggers auto meta creation. When you try to create a device greater than min_auto_meta_size and auto_meta is enabled then a meta will be created. Possible values are listed in Table 13 on page 229.
pav_alias_limit	If PAV is enabled, specifies the maximum number of aliases that can be assigned to a device. Possible values are: Enginuity 5671: 1-15 Enginuity 5771/5772: 1-255 Enginuity 5773 to 5876: 1-127

Table 9 Possible metric names (Enginuity 5773 through 5876)

PAV_mode	Enables the use of PAV (Parallel Access Volumes). Possible values are: STANDARD: Standard PAV volumes with static aliasing can be configured. DYNAMIC_STANDARD: Standard PAV volumes with dynamic aliasing can be configured.
rdfa_cache_percent	Sets the percentage of write pending cache that can be used by SRDF/A. This is a value from 0 to 100 percent.
rdfa_host_throttle_time	Sets the number of seconds to throttle host writes to SRDF/A devices when cache is full, before dropping RDF/A sessions. Throttling delays a write from the host until a cache slot becomes free. Values are from 0 to 65535. This option is not supported on VMAX 10K/VMAXe arrays.

Viewing the array metrics

You can view the current settings for these metrics with the following command:

```
symcfg -sid SymmID -v list
```

Managing external disk groups

Solutions Enabler includes a feature called Federated Tiered Storage (FTS). With FTS, an external LUN (eDisk) can be attached through the SAN to the array and can be used as external back-end disks for that array.

Refer to Chapter 9, "Federated Tiered Storage," for eDisk configuration details.

Creating devices (HYPERMAX OS 5977)

For arrays running HYPERMAX OS 5977, valid device configurations are:

- TDEV+BCV
- TDEV

When creating RDF device pairs, the following types are supported:

- RDF1+TDEV or RDF1-TDEV
- RDF2+TDEV or RDF2-TDEV

In a command file, you can create (add) one or more devices in a specified array. To create one or more devices, use the following command:

```
create dev count=<n>,
          size = <n> [MB | GB | CYL],
          emulation=<EmulationType>,
          config=<DevConfig>
          [, preallocate size = <ALL>
            [, allocate_type = PERSISTENT]]
          [, remote_config=<DevConfig>, ra_group=<n>]
          [, sg=<SgName> [, remote_sg=<SgName>]]
          [, mapping to dir <director_num:port>
            [starting] target = <scsi_target>,
            lun=<scsi_lun>, vbus=<fibre_vbus>
            [starting] base_address = <cuu_address>[,...]]
          [, device_attr =
```

```
<SCSI3_PERSIST_RESERV | DIF1 |
  AS400_GK>[,...]]
 [, device_name='<DeviceName>'[,number=<n | SYMDEV> ]];
```

Where:

`count` — Indicates the number of devices to create.

`size` — Specifies the size of the device needed in number of cylinders (default), megabytes, gigabytes, or terabytes.

Table 10 Maximum device sizes (HYPERMAX OS 5977)

	MBs	CYLs	GBs	TBs
FBA	16777216	8947848	16384	16

`emulation` — Specifies the device emulation type. Valid emulations are FBA, Celerra_FBA (external gateway), and AS/400_D910_099.

`config` — The device configuration type. Valid *`DevConfig`* types are TDEV and TDEV+BCV.

`preallocate size` — Indicates the amount of space pre-allocated to the thin device(s) when it is bound to a pool. The only valid value is ALL.

`allocate_type` — Indicates the allocation type. The only valid value is PERSISTENT.

`remote_config` — Specifies the desired remote RDF configuration (if any). Valid types when creating RDF pairs with a remote array running HYPERMAX OS 5977 are:

- RDF1+TDEV or RDF1-TDEV
- RDF2+TDEV or RDF2-TDEV
- RDF1-BCV+TDEV or RDF1-BCV-TDEV
- RDF2-BCV+TDEV or RDF2-BCV-TDEV

`ra_group` — Specifies the RA group number in the SRDF environment.

`sg` — The local array storage group name for devices involved in RDF operations.

`remote sg` — The remote array storage group name for devices involved in RDF operations.`s`.

`mapping to dir` — Specifies the director and port for the mapping operation.

For arrays running HYPERMAX OS 5977, `mapping to dir` returns a “feature not supported” error when the request includes mapping of devices to be created to ACLX enabled front end ports. In addition, the valid range for `port` is extended to 0 to 31.

`scsi_target` — Indicates a hex value for the SCSI target ID.

`scsi_lun` — Indicates a hex value for the SCSI logical unit number.

`vbus` — The virtual fibre bus address used when mapping to an FA port using volume set addressing.

`base_address` — Indicates a base or alias address for a device being mapped to an EA or EF port. These are mainframe ports which expect devices to be mapped in groups to form CU images.

`device_attr` — Specifies the attributes to be set on the new device. Possible values include:

- `SCSI3_persist_reserv` (persistent group reservation) You can set or clear the SCSI-3 persistence attribute only if the device is unmapped.
- `DIF1`
- `AS400_GK`

`device_name` — Specifies the user supplied name with a maximum of 64 characters including the suffix. Any character may be used for the device name except quotes (" "), which denote the start and end of input. The device name plus an optional suffix can have a maximum of 64 characters. If using a numerical suffix, the device name will be limited to 50 characters (prefix) and the trailing numerical suffix number will be limited to 14 characters. If not using a numerical suffix, all 64 characters can be specified for the device name. The maximum starting suffix is 1000000.

Note: The `device_name` parameter is supported only for Enginuity 5874 or higher.

`number` — Represents the user supplied number for the starting suffix. Specifying SYMDEV means that the corresponding device number will be used as the suffix.

Restrictions when setting device names

The following restrictions and limitations apply to setting device names:

- If setting a device name when creating RDF pairs, both RDF devices will be set to the same device name if using the SYMDEV option.
- Solutions Enabler does not check for uniqueness of device names.

The output displayed during the config change session shows the types of devices created.

Restrictions when creating devices

For arrays running HYPERMAX OS 5977, the following restrictions apply for creating and managing VMAX devices and storage pools:

- Arrays running HYPERMAX OS 5977 allows the creation externally-visible (TDEV) devices (up to 16TB), without requiring meta devices. The creation and control of meta devices is no longer supported. Array-wide meta settings, previously used to automatically create meta devices are not supported.

  Note: Requests to create FBA meta and CKD RAID-10 (ckd_meta) devices are blocked on these arrays.

- Creation of standard (disk group) provisioned devices is blocked. This includes internal devices that are used for the backing of thin pools (DATADEVs and SAVEDEVs) as well as Diskless and VDEV devices, which are implemented as disk group provisioned devices.
- Creation of gatekeeper devices is supported, but they will always be thin devices.
- The ACLX attribute is supported only for thin devices with FBA emulation.

- Encapsulation of user data as disk group provisioned devices is not supported.
- Partial allocation of thin devices is not supported starting with Solution Enabler 8.0.1.
- You cannot create or convert to SRDF device types when:
  - Domino mode is enabled on any current SRDF pairs.
  - There are any invalid tracks on any of the current SRDF devices.
  - Concurrent SRDF is enabled on a device.
- The only emulation type supported for iSeries devices is D910_099, as it is the only iSeries emulation type that supports thin provisioned devices. The size range is a minimum of 1562 cylinders and a maximum of 1118481 cylinders.
- AS/400_D910_099 thin devices cannot be CKD metadevices, SAVE devices, or DATA devices. In addition, IBM i thin devices cannot have the following attributes:
  - RCVRPNT_TAG
  - ACLX
  - DIF1

To create gatekeeper devices, use the `symconfigure create gatekeeper` command. All information about gatekeepers can be found in the *Solutions Enabler Installation Guide* and the EMC Knowledgebase solution EMC255976 available on EMC online support.

Note: Only thin gatekeeper devices can be created on arrays running HYPERMAX OS 5977. Both thin and standard gatekeeper devices can be created on arrays running Enginuity 5876 and lower.

Creating devices (Enginuity 5773 through 5876)

Creation of certain device (configuration) types via the `symconfigure` command may require two sessions. Table 11 describes the sessions required to create a device type and/or convert the various types of devices from one configuration type to another.

Table 11 Steps to create a device (page 1 of 3)

Desired device type	Session 1 create	Session 2
2-Way-BCV-Mir	2-Way-BCV-Mir	
2-Way-Mir	2-Way-Mir	
4-Way-Mir [a]	4-Way-Mir	
BCV	BCV	
BCV+TDEV	BCV+TDEV	
RDF1-BCV+TDEV	BCV+TDEV	Convert to RDF1-BCV+TDEV
RDF2-BCV+TDEV	BCV+TDEV	Convert to RDF2-BCV+TDEV
DRV[b]	DRV	

Table 11 Steps to create a device (page 2 of 3)

Desired device type	Session 1 create	Session 2
2-Way-DRV-Mir[f]	2-Way-DRV-Mir	
Parity RAID	RAID-S[c]	
RAID 5	RAID-5	
RAID 6	RAID-6	
RAID 5 BCV	BCV+R-5	
RAID 6 BCV	BCV+R-6	
RDF1	RDF1	
RDF1+R-5	RDF1+R-5	
RDF1+R-6	RDF1+R-6	
RDF1-BCV	BCV	Convert to RDF1-BCV
RDF1-BCV+R-5	RDF1+R-5	Convert to RDF1-BCV+R-5
RDF1-BCV+R-6	RDF1+R-6	RDF1-BCV+R-6
RDF1-BCV-Mir	RDF1+Mir	RDF1-BCV-Mir
RDF1-Mir	RDF1-Mir	
RDF2	RDF2	
RDF2+R-5	RDF2+R-5	
RDF2+R-6	RDF2+R-6	
RDF2-BCV	BCV	RDF2-BCV
RDF2-BCV+R-5	RDF2+R-5	RDF2-BCV+R-5
RDF2-BCV+R-6	RDF2+R-6	RDF2-BCV+R-6
RDF2-BCV-Mir	RDF2+Mir	RDF2-BCV-Mir
RDF2-Mir	RDF2-Mir	
RDF21[d]+R5	RDF1+R5	Add R2 mirror
RDF21[d,e]+R5	RDF2+R5	Add R1 mirror
RDF21[d,e]+R6	RDF1+R6	Add R2 mirror
RDF21[d,e]+R6	RDF2+R6	Add R1 mirror
RDF21[d,e]-Mir	RDF2-Mir	Add R1 mirror
RDF22[e,f]+R5	RDF2+R5	Add R2 mirror
RDF22[e f] +R6	RDF2+R6	Add R1 mirror
RDF1+TDEV	RDF1+TDEV	
RDF2+TDEV	RDF2+TDEV	
RDF1-BCV+TDEV	RDF1+TDEV	Convert to RDF1-BCV+TDEV

Table 11 Steps to create a device (page 3 of 3)

Desired device type	Session 1 create	Session 2
RDF2-BCV+TDEV	RDF2+TDEV	RDF2-BCV+TDEV
DLDEV[f]		
RDF1+DLDEV[f]		
RDF2+DLDEV[f]		
RDF21+DLDEV[f]	RDF2+DLDEV	AddR1 mirror
RDF21+DLDEV[f]	RDF1+DLDEV	AddR2 mirror
Thin device (TDEV)[d]	TDEV	
Unprotected	Unprotected	
Virtual	VDEV	

a. 4-Way-Mir devices cannot have any BCV devices. 4-Way-Mir devices are not supported on Enginuity 5874 through 5876.

b. Not supported on Enginuity 5874 and higher.

c. Not allowed for Enginuity 5771 and higher.

d. Only for Enginuity 5773 and higher.

e. To create an R21or R22 device, use add rdf mirror with the symconfigure command. Note that this command will fail if the device is a Data Domain device.

f. Only for Enginuity 5874 and higher.

For information on creating dynamic-RDF capable devices, refer to "Configuring dynamic RDF-capable devices (Enginuity 5773)" on page 262.

When you create an SRDF device where a corresponding device must be created on a remote SRDF-linked array, a configuration change session is initiated and managed on the remote array by the local `symconfigure` utility.

Note: SAVE devices and DATA devices are created following the same procedure as for standard protected devices, with the additional specification of `attribute=savedev` or `attribute=datadev`.

In a command file, you can create (add) one or more devices in a specified array. To create one or more devices, use the following command:

```
create dev count=n,
size=n [MB |GB | CYL],
emulation=EmulationType,
config=DeviceConfig,
[, data_member_count=nn]
[, remote_config=DeviceConfig,
   remote_data_member_count=nn,
   ra_group=n, [remote_mvs_ssid=nnn],
   [dynamic_capability=[dyn_rdf | dyn_rdf1_only | dyn_rdf2_only]]
[, mvs_ssid=nnn]
[, attribute=ckd_meta | savedev | datadev
  [in pool PoolName] [member_state=ENABLE | DISABLE] ]
[meta_member_size = <n> [MB | GB | CYL]]
[meta_config = [striped | concatenated]][, disk_group=nnn|
name:DskGrpName,
[remote_disk_group=nnn | name:DskGrpName] ],
```

```
    [, binding to pool=PoolName
    [preallocate size = <ALL | n [ MB | GB | CYL]>]
    [allocate_type = persistent]
    [remote_pool=PoolName]]]
[, [mapping to dir DirNum:PortNum
   [starting] target = scsi_target,
    lun=scsi_lun, vbus=fibre_vbus
    [starting] base_address= <cuu_address>]...]
[[device_attr =
   [SCSI3_persist_reserv | ACLX |DIF1 |AS400_GK]]...]
[,device_name=DeviceName
   [,number=<n | SYMDEV>]];
```

Where:

`count` — Indicates the number of devices to create.

`size` — Specifies the size of the device needed in number of cylinders (default), megabytes, or gigabytes.

Table 12 Maximum device sizes (Enginuity 5773 through 5876)

Enginuity version	MBs	CYLs	GBs
Enginuity 5874 through 5876	245760	262668	240
Enginuity 5773 and lower	61425	65520	59

To calculate the number of cylinders, use the following:

1 cylinder = 15 tracks; each track is 64 KB

15 x 64 tracks =.937 MB for each cylinder

Note: Devices to be used as BCV, RDF, or metamembers will need to precisely match corresponding device sizes. Use `symdev/sympd show` or `symdev/sympd list -cyl` commands to see relevant device sizes.

Sizing limits for CKD devices are:

CKD 3380 — 3393 cyls

CKD 3390 — 262668 cyls

`emulation` — Specifies the device emulation type. With Solutions Enabler and Enginuity 5875 and 5876, you can create an AS/400 volume of any size and meta configuration for the following emulation types:

`AS/400_M2107_099` — Specifies a volume type 2107_A99 to be treated as protected (equivalent to the current A0*x* models).

`AS/400_M2107_050` — Specifies a volume type 2107_A50 to be treated as unprotected (equivalent to the current A8*x* models).

`AS/400_D910_099` — Specifies a volume type D910_099. These devices cannot be disk group provisioned devices; the command should always specify TDEV. The protection types supported are TDEV, RDF_R1_TDEV, RDF_R2_TDEV, and BCV_TDEV. Refer to Table 16 for the protection types supported when converting these devices.

If `auto_meta` is enabled, the command follows the `auto_meta` rules for non-AS400 FBA devices to create metadevices. Also, similar to non AS/400 FBA devices, the `meta_member_size` and `meta_config` can be used to override the `auto_meta` settings on the system.

Note: The protection interpretation by the OS is not related to the actual array RAID protection.

`config` — Specifies the desired device configuration type. Possible values are shown in the second column of Table 11 on page 217.

`data_member_count` — Indicates the number of data members when creating a RAID 5 or RAID 6 devices on an array with Enginuity version 5772 through 5876.

Value for data_member_count	Protection Type
3	RAID5 (3+1)
7	RAID5 (7+1)
6	RAID6 (6+2)
14	RAID6 (14+2)

`remote_config` — Specifies the desired remote SRDF configuration (if any) from Table 11 on page 217.

`remote_data_member_count` — Specifies the number of remote data members when creating SRDF RAID 5 or RAID 6 devices on an array with Enginuity version 5772 through 5876. You should set the value to 3 or 7 for RAID 5 (3+1) and RAID 5 (7+1) or, 6 or 14 for RAID 6 (6+2) and RAID 6 (14+2).

`ra_group` — Specifies the RA group number in the SRDF environment.

`remote_mvs_ssid` — When creating an SRDF device in a remote array that also contains CKD devices, a z/OS (MVS) subsystem ID (`remote_mvs_ssid`) value may be provided so the new FBA devices are not seen as part of an existing subsystem ID group. Only one `mvs_ssid` and `remote_mvs_ssid` can be used in a session. They will be applied to all devices created within that session.

`dynamic_capability` — Specifies the type of dynamic SRDF device to create. Possible values are:

- `dyn_rdf` — Creates a dynamic SRDF device.
- `dyn_rdf1_only` — Creates a dynamic R1 SRDF device.
- `dyn_rdf2_only` — Creates dynamic R2 SRDF device.

Note: All devices on VMAX 10K Series systems are RDF-capable, therefore the `dynamic_capability` option is disabled on those systems.

With Enginuity 5874 and 5876, a single device can be set to be dynamic without automatically creating a device pair.

Note: Enginuity 5874 Q2 2011SR supports creating thin BCV devices (BCV+TDEV and BCV-TDEV) with the dynamic capability set (`dyn_rdf`, `dyn_rdf1_only`, and `dyn_rdf2_only`. These devices can later be converted to be thin SRDF BCV devices. Refer to "Converting devices" on page 236.

When creating a dynamic pair, the specified dynamic capability is applied to the local device, and the corresponding remote device is assigned a complementary dynamic capability, according to the following:

Local device	Remote device
dyn_rdf	dyn_rdf
dyn_rdf1_only	dyn_rdf2_only
dyn_rdf2_only	dyn_rdf1_only

`mvs_ssid` — When creating a device in a array that also contains CKD devices, a z/OS (MVS) subsystem ID (`mvs_ssid`) value can be provided so the new FBA devices are not seen as part of an existing subsystem ID group. For more information about SSIDs, refer to "Creating devices (Enginuity 5773 through 5876)" on page 217 or "Creating devices (HYPERMAX OS 5977)" on page 214.

Note: The `mvs_ssid` and `remote_mvs_ssid` parameters are optional when creating FBA devices on arrays that have FICON or ESCON adapters.

`attribute` — Specifies the desired device attribute. Possible values are:

`ckd_meta` — When creating a device with emulation type of CKD-3390, this indicates that the device should be a striped metadevice. CKD metadevices must be created in sets of four devices.

`savedev` — When creating a device, this indicates that the device should be a SAVE device. The device will become part of a pool of devices that are used with TimeFinder/Snap for virtual device Snap operations.

`datadev` — When creating a device, this indicates that the device should be a DATA device. The device will become part of a pool of devices that are used with thin devices for virtual operations.

Note: If the array is running Enginuity 5874 or higher, a disk group must be specified when creating a SAVE device or DATA device if the `create dev` command is also adding the device(s) to a pool.

`PoolName` — Specifies the name of the SAVE or thin pool. It can be from 1 to 12 alphanumeric characters long and include hyphens (-), and underscore (_) characters. The name `DEFAULT_POOL` is reserved to represent the container of all *unpooled* SAVE devices. Thin pools do not use the `DEFAULT_POOL`.

Note: The pool name `DF_DDEV_POOL` is reserved for EMC use only.

`member_state` — Indicates whether the devices being added should be enabled or disabled in the pool.

Note: The auto meta setting is enabled array-wide by default. Metadevices can be created automatically by using the `meta_member_size` and the `meta_config` options.

`meta_member_size` — Sets the size for the creation of metadevices.

`meta_config` — Sets the metadevice configuration to striped or concatenated.

`disk_group` — Specifies a disk group number (or name) to which the created device belongs. A disk group is a set of physical disks set aside to be used to create devices of a specific protection level. Disk group usage may improve the performance of some configurations.

`name` — Specifies the name of the disk group. By default, the disk group name is `DISK_GROUP_`*`xxx`*, where *`xxx`* is the disk group number.

Usage is: `disk_group=#` or `disk_group=name:`*`DskGrpName`*

`remote_disk_group` — Specifies a remote disk group number (or name) when creating a device.

`name` — Specifies the name of the remote disk group. By default, the disk group name is `DISK_GROUP_`*`xxx`*, where *`xxx`* is the disk group number.

`binding to pool` — Specifies the thin *PoolName*. This option is only available when creating thin devices. When creating SRDF thin devices, both SRDF devices must be bound at the same time. Individual RDF1 and RDF2 devices cannot be bound separately. Devices are not required to be bound to a pool upon creation, however, they cannot be used until they are bound to a pool.

`preallocate size` — Indicates the amount of space (pre) allocated to the thin device(s) when it is bound to a pool. The only valid value is ALL for all Enginuity levels.

`remote_pool` — Specifies the name of the remote thin pool. This option is only for SRDF thin devices.

`mapping to dir` — Specifies the director and port for the mapping operation.

`scsi_target` — is a hex value for the SCSI target ID.

scsi_lun — Indicates a hex value for the SCSI logical unit number.

`fibre_vbus` — The virtual bus address used when mapping to an FA port using volume set addressing.

`cuu_address` — Indicates a base or alias address for a device being mapped to an EA or EF port. These are mainframe ports which expect devices to be mapped in groups to form CU images.

`device_attr` — Specifies the attributes to be set on the new device. Possible values include:

- `RDB_Cksum` (Enginuity 5773 and lower)
- `VCMdb` (for device masking on Enginuity 5773 and lower)

- `SCSI3_persist_reserv` (persistent group reservation) In Enginuity 5874 and lower, this attribute is disabled by default. In Enginuity 5875 and 5876, this attribute is enabled by default. When running Enginuity 5876 Q42012SR, you can set or clear the SCSI-3 persistence attribute only if the device is unmapped.
- `dyn_rdf` (This option provides the most flexibility in performing dynamic SRDF operations)
- dyn_rdf1_only
- dyn_rdf2_only
- `ACLX` (For Auto-Provisioning Groups on Enginuity 5874 and 5876) This attribute is not allowed on IBM i thin devices.
- `RCVRPNT_TAG` (for RecoverPoint with Enginuity 5875 Q2 2011 SR and 5876)
- `DIF1` (Enginuity 5876)
- `AS400_GK` (Enginuity 5876)

Note: Refer to "Setting device attributes" on page 241 for a list of the device attribute restrictions.

`device_name` — Specifies the user supplied name with a maximum of 64 characters including the suffix. Any character may be used for the device name except quotes (" "), which denote the start and end of input. The device name plus an optional suffix can have a maximum of 64 characters. If using a numerical suffix, the device name will be limited to 50 characters (prefix) and the trailing numerical suffix number will be limited to 14 characters. If not using a numerical suffix, all 64 characters can be specified for the device name. The maximum starting suffix is 1000000.

Note: The `device_name` parameter is supported only for Enginuity 5874 or higher.

`number` — Represents the user supplied number for the starting suffix. Specifying `SYMDEV` means that the corresponding device number will be used as the suffix.

Restrictions when setting device names

The following restrictions and limitations apply to setting device names:

- If setting a device name when creating RDF pairs, both RDF devices will be set to the same device name if using the SYMDEV option.
- Solutions Enabler does not check for uniqueness of device names.
- Device names for the following devices are not supported: SFS, DRV, metamembers, SAVE, DATA, Vault, and diskless devices.

The output displayed during the config change session shows the types of devices created.

Restrictions when creating devices

- There are restrictions on the use of unprotected devices created by `symconfigure`, or converted to unprotected devices. Such devices cannot be mapped to hosts for normal data storage use. If an unprotected device is intended for use as a gatekeeper, it should be created with 48 cylinders or less in order to be mapped to a host.
- You can create devices with a variety of emulation types: Fixed Block Architecture (FBA), Celerra FBA or VME512 FBA, and CKD for z/OS environments.

- You cannot create or convert to SRDF device types when:
  - Domino mode is enabled on any current SRDF pairs.
  - There are any invalid tracks on any of the current SRDF devices.
  - Concurrent SRDF is enabled on a device.
- For AS/400 types, 2107_A99 and 2107_A50, a size must be specified. The size range allowed for these devices is a minimum of 165 cylinders and a maximum of 2236963 cylinders. For AS/400 type D910_099 (supported with Solutions Enabler V7.6 and higher) the size range is a minimum of 3052 cylinders and a maximum of 2236962 cylinders.
- An AS/400 device or a device tagged for RecoverPoint cannot be mapped to an FCoE director.
- AS/400_D910_099 thin devices cannot be CKD metadevices, SAVE devices, or DATA devices. In addition, IBM i thin devices cannot have the following attributes:
  - RCVRPNT_TAG
  - ACLX
  - DIF1
- Partial allocation of thin devices is not supported starting with Solution Enabler 8.0.1.

When creating devices in an z/OS environment, all new devices must have the same MVS SSID and any remote devices must also have the same remote MVS SSID within a single session. To create devices with different SSID values, you will need to run more than one session.

To create gatekeeper devices, use the `symconfigure create gatekeeper` command. All information about gatekeepers can be found in the *EMC Solutions Enabler Installation Guide* and the EMC Knowledgebase solution EMC255976 available on EMC online support.

Note: Only thin gatekeeper devices can be created on arrays running HYPERMAX OS 5977. Both thin and standard devices can be created on arrays running Enginuity 5876 and lower.

SRDF device pairs — With Enginuity 5876, when you create an RDF device pair and do not provide the `mapping to dir` option, the devices are RW on the link and are marked synchronized immediately. This means that no data copy occurs. The devices immediately report a Synchronized RDF pair state.

SRDF mode default — The SRDF mode on an SRDF device pairing will be Adaptive Copy Disk by default, unless the SYMAPI_DEFAULT_RDF_MODE is set in the option file. If the device being created is a diskless R1 device, the RDF mode will default to Adaptive Copy Write Pending, regardless of the option file setting. (Applies to arrays running Enginuity 5876.)

VDEV configuration restrictions — Virtual devices are configured as full-size devices, but use minimal disk storage on the back-end. On DMX arrays, VDEVs reside in cache only and are depicted as having no back-end hypers. The format must be FBA. Virtual devices can also be used to form a VDEV metadevice. Other devices cannot be converted to form virtual devices. To configure a virtual device, set `config=vdev` in the device file.

SAVE device configuration restrictions — SAVE devices work in conjunction with virtual devices and can be unprotected, mirrored, or RAID 5/RAID 6. Configuring SAVE devices as unprotected is not recommended and requires special approval by EMC. SAVE devices cannot be part of a metavolume or grouped. For a given device emulation type, all SAVE devices must have the same protection. SAVE devices cannot be converted to other devices. Configuring virtual and SAVE devices requires careful planning to anticipate what percentage of the data may change during a virtual copy session. Creating the SAVE device pool too small could result in data changes being lost during a copy session. Make sure that you create enough SAVE devices and set the size of the SAVE devices large enough to handle any expected changes. To configure a SAVE device, set `attribute=savedev` in the request to create a new device.

Copying devices

You can configure available disk space into a set of new devices by copying the attributes of an existing device. To do this, either specify the quantity of disk space to configure into the new devices, or the number of devices to be created. The quantity of disk space represents the new space that will be available for the host to use, and not the space allocated in the array to manage the request according to the device protection requirements. In addition, you can also override some of the copied device's attributes in the new devices. For example, you can copy the attributes of a standard device, but configure the new devices as BCVs.

To configure a device by copying a similar device, use the following form:

```
configure [n.nn [MB | GB] | nn devices]
copying dev SymDevName
[mapping to dir DirectorNum:PortNum
 [masking hba [awwn=awwn | wwn=wwn |
                iscsi=iscsi | aiscsi=aiscsi]
                   host_lun=lun |dynamic_lun]] ]
[,device_name=DeviceName
  [,number=n | SYMDEV]]
[overriding
                    [size=<n> [MB | GB | CYL]]
                    [emulation=<EmulationType>]
                    [config=<DevConfig>]
                    [data_member_count=<n>]
                    [mvs_ssid=<n>]
                    [disk_group=<n>| name:<DskGrpName>]];
```

Note: For arrays running HYPERMAX OS 5977, the configure operation is blocked if the source device specified by *SymDevName* is a TDAT device.

Where:

n.nn `[MB | GB]` — Specifies the quantity of disk space to configure.

nn `devices` — Specifies the number of devices to create.

SymDevName — The device name of the model device.

DirectorNum:*PortNum* — The mapping attributes of the device are not copied. If the new devices are to be mapped, you must specify the director/port addresses.

`mapping to dir` — Specifies the director and port for the mapping operation.

For arrays running HYPERMAX OS 5977, `mapping to dir` returns a feature not supported error when the request includes mapping of devices to be created to ACLX enabled front end ports. In addition, the valid range for *`PortNum`* is extended to 0 to 31.

`masking hba` — The masking attributes of the model device are not copied. If the new devices are to be masked, you must specify the host HBA to which the devices should be masked. This option is not available on arrays running Enginuity 5874.

`host_lun` — Specifies the LUN addresses to be used for each device that is to be added for the host HBA.

`dynamic_lun` — Specifies to use the dynamic LUN addressing features but does not require a LUN address for each device. The LUN addresses are assigned based on what may already be in use for that host HBA.

`device_name` — Specifies the user supplied name with a maximum of 64 characters including the suffix. The legal characters for the device name include all alpha, numeric, underscore(_) and period(.). The device name plus an optional suffix can have a maximum of 64 characters. If using a numerical suffix, the device name will be limited to 50 characters (prefix) and the trailing numerical suffix number will be limited to 14 characters. If not using a numerical suffix, all 64 characters can be specified for the device name. The maximum starting suffix is 1000000.

Note: The `device_name` parameter is supported only for Enginuity 5874 or higher.

`number` — Represents the user supplied number for the starting suffix. Specifying `SYMDEV` will mean that the corresponding device number will be used as the suffix.

`overriding` — Indicates that you will be overriding some of the characteristics of the copied device.

`size` — Specifies the size of the new devices.

`emulation` — Specifies the device emulation type.

For arrays running HYPERMAX OS 5977, valid emulations are FBA and AS/400_D910_099.

For arrays running Enginuity 5773 through 5876, valid emulation types are listed on page 220.

`config` — The device configuration type.

For arrays running HYPERMAX OS 5977, valid configuration types are TDEV and TDEV+BCV.

For arrays running Enginuity 5773 through 5876, possible configuration types are listed on Table 11 on page 217.

`data_member_count` — For arrays running Enginuity 5773 through 5876, specifies the number of data members when creating a RAID 5 or RAID 6 devices on an array with Enginuity version 5772 through 5876. You should set the value to 3 or 7 for RAID 5 (3+1) and RAID 5 (7+1) or, 6 or 14 for RAID 6 (6+2) and RAID 6 (14+2). This option is not valid on arrays running HYPERMAX OS 5977.

`mvs_ssid` — When creating devices in an array that also contains CKD devices, a z/OS (MVS) subsystem ID (`mvs_ssid`) value can be provided so the new FBA devices are not seen as part of an existing subsystem ID group.

`disk_group` — For arrays running Enginuity 5773 through 5876, specifies the disk group number (or name) in which to place the new devices. If this option is not specified, the model device's disk group is *not* used. Instead, the system will place the hypers on any available disk. This option is not valid on arrays running HYPERMAX OS 5977.

`name` — Specifies the name of the disk group. By default, the disk group name is `DISK_GROUP_`*`xxx`*, where *`xxx`* is the disk group number.

Usage is: `disk_group=`**#** or `disk_group=name:`*`DskGrpName`*

Managing metadevices (Enginuity 5773 through 5876)

Note: Solutions Enabler V8.0.1 does not support the use of metadevices with HYPERMAX OS 5977.

Metadevices allow individual devices to be concatenated to create larger devices. The devices assigned in a meta sequence do not need to be adjacent.

The *metahead* is the first device in the metadevice and is responsible for receiving incoming commands. When an incoming command for the metahead is processed, the Enginuity software determines which metadevice member should execute the command.

Several operating systems, such as Windows and some open systems environments, can have larger volumes than are provided by standard array physical disk devices. A metadevice is a logical volume set created from individual array hypers to define volumes larger than the current array maximum hyper volume size of approximately 16 GB. Metadevices are functionally the same as logical volume sets implemented with the host volume manager software. Physically, a metadevice is two or more volumes presented to a host as a single addressable device. It consists of a metahead device, some number of member devices (optional), and a metatail device (see Figure 6).

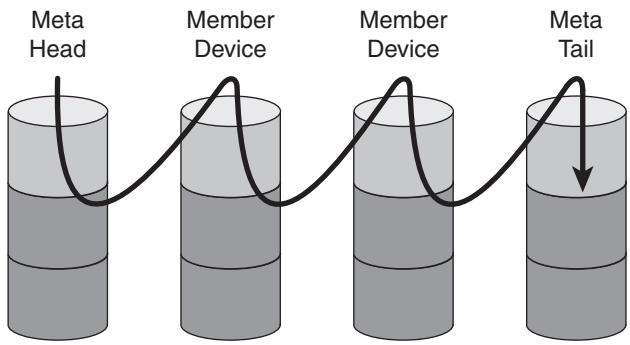


Figure 6 Concatenated metadevices

Metadevice size requirements

Metadevices may contain as many as 255 members. The maximum size metadevice that has been tested by EMC is 60 terabytes, which is the maximum size supported. Metadevices can consist of nonsequential and nonadjacent volumes.

Table 13 Metadevice sizes by Enginuity version for auto_meta

Enginuity version	Max single device size (CYL)	Max single device size (GB)	Min_auto_meta_size	Auto_meta_member_size
Enginuity 5874 and 5876	262668	240	262669	262668
Enginuity 5773	65520	59	65521	65520

Meta performance

By allowing individual physical disk devices to be grouped together into a metadevice and the capability of metadevice addressing, the array enhances disk system functionality. To increase throughput and further improve performance, the array provides multiple I/O drive queues for metavolumes.

Accessing data in a metadevice

Addressing of data contained in a metadevice can be organized in two different ways:

- Concatenated devices
- Striped devices

Concatenated devices

Concatenated devices are volume sets that are organized with the first byte of data at the beginning of the first device (Figure 6 on page 228). Addressing continues to the end of the first device before any data on the next device is referenced. When writing to a concatenated device, the first metadevice member receives all the data until it is full, and then data is directed to the next member and so on.

Striped devices

Metadevice addressing by striping divides each metamember device into a series of stripes, addressing a stripe from each device before advancing to the next stripe on the first device (as shown in Figure 7). When writing to a striped volume, equal size stripes of data from each participating drive are written alternately to each member of the set.

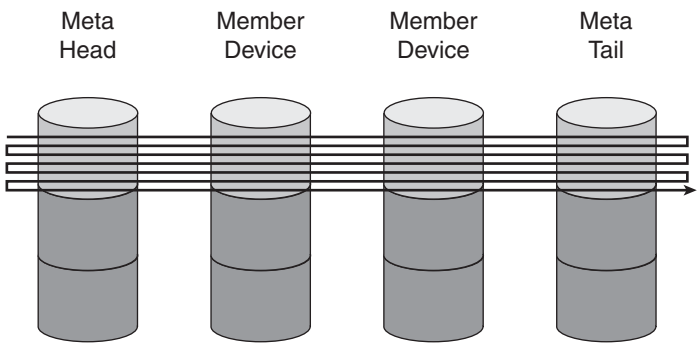


Figure 7 Striped data

Striping data across the multiple drives in definable cylinder stripes benefits *sequential writes* by avoiding stacking multiple writes to a single spindle and disk director. This scheme creates a large volume, but additionally balances the I/O activity between the disk devices and the array disk directors.

If no stripe size is specified when creating a striped meta, all Enginuity codes will consider the default stripe size as 1920 blocks of 512 bytes each:

- 1 cylinder in Enginuity 5671 and earlier = 960 512-byte blocks
- 1 cylinder in Enginuity 5773 and higher = 1920 512-byte blocks

Note: The sum of the metamember's individual cylinder size and the metahead cylinder size may not match. A `symdev show` of the metahead device only shows the usable cylinder size.

Creating metadevices

Metadevices can be created in either of the following ways:

- Automatically, with the create device command, as explained in "Creating devices (Enginuity 5773 through 5876)" on page 217.

  Note: The auto meta setting is enabled by default. Metadevices can be created automatically by using the `meta_member_size` and the `meta_config` options.

- After creating a device, as explained in "Forming metadevices" on page 230.

Forming metadevices

The automatic creation of metadevices can be accomplished with one configuration change session. Metadevice configurations that require two sessions (`form` and `convert`) are shown in Table 14.

Table 14 Steps to create an SRDF BCV metadevice

Desired device type	Session 1 form	Session 2 convert
Meta RDF1-BCV	Meta-BCV	→ Meta RDF1-BCV
Meta RDF2-BCV	Meta-BCV	→ Meta RDF2-BCV

If you do not use the automatic meta creation feature, you can form a metadevice using the `form meta` command.

Note: Cacheless Read Miss support can return an error if the specified devices used to form a meta contain different Cacheless Read Miss modes.

When you initially create a metadevice, you must form the head and add the members in the same command file. If using the `count` option, no members need to be specified. The configuration server will automatically select members from the pool of unmapped devices that match the metahead in size, emulation, and configuration type.

Note: Make sure none of the devices being considered for a metadevice have the SCSI-3 persistent reserve flag enabled.

To form a metadevice, use the following command:

```
form meta from dev SymDevName,
config=MetaOption
[, stripe_size=<MetaStripeSize>[cyl]]
[, count=<member_count>];
```

where:

`config` — Specifies the meta configuration type. Possible values are `concatenated` or `striped`.

`stripe_size` — Specifies the size of the striped metadevice. This value can be expressed in blocks or cylinders. Possible sizes in 512 byte blocks are 1920, 3840, 7680, 15360, 30720, and 61440. The stripe size must be 1920, which is the default for all versions of Enginuity. If no stripe size is specified when creating a striped meta, all Enginuity codes will consider the default stripe size as 1920 blocks of 512 bytes each. For information on metavolume stripe sizes, refer to "Striped devices" on page 229.

`cyl` — Indicates that the `stripe_size` is expressed in cylinders. The size of a cylinder for FBA emulation is 960 512-byte blocks.

`count` — Indicates the total number of devices for the configuration server to add to the new metadevice, including the head. This option may be omitted if you want to specify the members using the `add dev` command.

RDF metadevices

The `create dev` command syntax also can create static and dynamic SRDF metadevices. To use this feature, both the local and the remote arrays must be running Enginuity 5773 through 5876 and the local array must have `auto_meta` enabled. The `auto_meta` feature may or may not be enabled on the remote array. However, any `auto_meta` settings on the remote array are ignored.

Assume that the local array is enabled for `auto_meta` and `min_auto_meta_size` is set to 30000. The following command file will create a 2-member metadevice, on both the local and the remote arrays:

```
create dev count=1, size = 60000, emulation=FBA, config=RDF1,ra_group=20,
remote_config=RDF2,dynamic_capability=dyn_rdf;
```

Note that SRDF devices cannot be used to form metadevices in one step. When this is necessary, the SRDF devices must first be converted to non-RDF devices (as shown in Table 14), the metadevices must be formed separately on both arrays, and then the metahead on one array can be converted back to an SRDF device.

Also, once the SRDF metadevice has been established, it cannot be modified in one step. If any modifications are required, it will again have to be converted to a non SRDF device first.

AS/400 metadevices

Many AS/400 devices are a fixed size. The emulation type determines the size of the device. Solutions Enabler determines the size automatically from the emulation type.

When `auto_meta` is enabled, Solutions Enabler attempts to create an AS/400 non metadevice and an AS/400 metadevice with or without a size value entered.

When `auto_meta` is disabled, Solutions Enabler will attempt to create an AS/400 non metadevice, but will reject a request to create an AS/400 metadevice.

For example, the following command creates an AS/400 metadevice on an array with emulation AS/400_M4328_50.

```
create dev count=1, emulation= AS/400_M4328_50, config=2-Way-Mir;
```

This will create an 8-member metadevice where total cylinder size is 143576 (native mode).

Meta configuration restrictions

Solutions Enabler no longer requires that non metadevices be unmapped while being converted to a concatenated metadevice. A device being converted to a concatenated metadevice can be mapped and online to the host during the operation.

Note: A non metadevice being converted to a striped metadevice must still be unmapped before it can be converted.

The following are restrictions when configuring metadevices:

- Metamembers cannot be removed from a striped metadevice.
- To create a striped metadevice, all members must be the same size and have the same mirror protection.
- To remove a member from a concatenated metadevice, the member must be the tail. You cannot remove inner members.
- Only the head of a meta may have its type converted.
- Only the head of a meta is mapped or assigned to the host.
- All metas must contain at least one metamember. When a meta is formed, at least one member must be added.
- All metadevices must be composed of devices of the same BCV type (you must convert from SRDF devices in order to change metas).
- All metamembers must belong to the same physical disk group.
- Metadevices must be composed of devices that are all of the same FBA emulation type.
- You must create CKD metadevices using the `create dev` command, not the `form dev` command.
- VDEV metadevices must be composed of virtual devices that are of the same capacity, stripe size, count, and emulation type as the standard devices with which they will be used.
- An SRDF device cannot be used to form a metadevice (refer to Table 14 on page 230 for converting a metadevice for SRDF operation). The configuration manager cannot form a metadevice from SRDF devices that are RDFA-protected (that is, you cannot convert these devices to non-RDF, form the metadevice, and convert them back to RDF).

Adding members to a concatenated meta

To add additional members to an existing concatenated metadevice, use the following form:

```
add dev SymDevName[:SymDevName] to meta SymDevName;
```

For example, to add devices 00013 and 00014 to concatenated meta 00010, enter:

```
add dev 00013:00014 to meta 00010;
```

The `-protect_data` option is not supported for concatenated metadevices.

Note: Cacheless Read Miss support can return an error if the specified devices added to a meta contain different Cacheless Read Miss modes.

Adding members to a striped meta

Solutions Enabler supports expanding a striped thin metadevice with data protection. To add additional members to an existing striped metadevice, use the following form:

```
add dev SymDevName[:SymDevName] to meta SymDevName
[, protect_data=[TRUE | FALSE],
 bcv_meta_head=SymDevName];
```

Where:

`protect_data` — Possible values are `TRUE` or `FALSE`. The `protect_data` option is only for striped metas. When set to `TRUE`, the configuration manager automatically creates a protective copy to the BCV meta of the original device striping. Because this occurs automatically, there is no need to perform a BCV establish. When set to `FALSE`, the data will not be preserved during the meta expansion.

Note: For arrays running Enginuity V5773 and earlier, RAID 5 BCVs are not supported for the `protect_data` option. You can protect data by using a mirrored BCV device.

`bcv_meta_head` — When adding new members to an existing, striped metadevice, if the data on the metadevice is to be protected, you must specify the name of a `bcv_meta_head` that matches the original metadevice in capacity, stripe count, and stripe size.

Note: Thin BCV devices can be used as the `bcv_meta_head`. The thin BCV metahead and all members must be bound to a thin pool.

For example, to add devices `00013` and `00014` to striped meta `00010`, enter:

```
add dev 00013:00014 to meta 00010,
protect_data=TRUE,
bcv_meta_head=00CA;
```

Removing metamembers

For example, to remove a metamember or members from the tail of a concatenated metadevice, use the following form:

```
remove dev SymDevName[:SymDevName] from meta SymDevName;
```

To remove device 00014 from meta 00010, enter the following in a command file:

```
remove dev 00014 from meta 00010;
```

Dissolving a metadevice

The `symconfigure` command file can use a list of device numbers or ranges for the `dissolve meta` command. This allows multiple metadevices to be dissolved with one command.

To dissolve multiple metas, provide a list with the device numbers of the metahead devices, or use the device numbers of the meta heads within a device range. The command will skip over the device numbers of metamembers found in the range, dissolving only metas for the metaheads found within the range. If non metadevices are found within the range an error will return.

To dissolve one or more metadevices, use the following form:

```
dissolve meta dev SymDevName [:SymDevName]
     [[,SymDevName[:SymdevName]]...];
```

For example, to dissolve multiple metadevices where the device numbers for the metas are contiguous from `d0` to `df`, including all the metaheads, create a command file containing the following command:

```
dissolve meta dev d0:df;
```

To dissolve the meta devices, use the following command:

```
symconfigure -sid 266 commit -file dissolve_meta.cmd -v -noprompt
```

Note: A `dissolve` action removes the metahead and all its members.

Data preservation concerns

When managing metadevices, data is not written directly to the devices involved, and therefore the original data is essentially preserved. However, due to the reorganization upon completion of the device management actions, the host is unable to access all of the original data. Therefore, it is the responsibility of the attached host to handle the change to the physical geometry of the device.

Note: The expansion of striped metadevices is an exception to this rule. In this case, you can provide a BCV meta to preserve the original data, which will be rewritten to the newly configured striped metadevice.

Table 15 lists the possible data management operations and their effects on data integrity.

Table 15 Metadevice data preservation

Metadevice type with data	Operation	Data integrity preserved status
Metadevices concatenated	Adding a member	Preserved
Metadevices concatenated	Removing a member	Preserved[a]
Metadevices concatenated	Dissolving to non meta	**Not** preserved[b]

Table 15 Metadevice data preservation

Metadevice type with data	Operation	Data integrity preserved status
Metadevices concatenated	Converting to striped	Configurable[c]
Metadevices striped	Adding a member	Configurable[c]
Metadevice striped	Dissolve	**Not** preserved[b]

a. Preserved up to where the metadevice is removed.
b. Preserved but there is no host component to piece the data together.
c. Protection is a user-configurable option.

CAUTION

Dissolving striped or concatenated devices cause data access loss.
Adding to concatenated metadevices preserves the metadevice data, but makes the new members's original data inaccessible.
Removing concatenated members cause data access loss on the removed member. Use these commands with great caution.

Configuring thin metadevices (Enginuity 5773 through 5876)

Note: Solutions Enabler V8.0.1 does not support the use of thin metadevices with HYPERMAX OS 5977.

The `create dev`, `form meta`, `dissolve meta`, `convert meta` commands can be used to configure a thin metadevice. Follow these guidelines when creating thin metadevices.

For a striped metadevice:

- To form a striped metadevice, all members must be unmapped, unbound, and have the same device size.

For a concatenated metadevice:

- On arrays running Enginuity versions lower than 5875, only unmapped thin devices can formed into metadevices.
- On arrays running Enginuity 5875 or 5876, bound thin devices can be used as metaheads; however, bound thin devices cannot be used as metamembers.
- On arrays running Enginuity 5875 or 5876, mapped or unmapped thin devices can be formed into concatenated metadevices.

Only thin devices can be used to create a thin metadevice. Mixing of thin and non thin devices to create a meta is not allowed.

Dissolving a thin meta

Only unbound thin metadevices can be dissolved.

Using thin BCV metadevices

Solutions Enabler with Enginuity 5875.198.148 and 5876 supports the use of a thin BCV metavolume while expanding striped thin metavolumes. The expansion guidelines are:

- All existing rules for using regular BCV devices apply to thin BCV (such as size, number of members, and member size).
- A thin BCV device used for data protection must be bound to a thin pool.
- A thin BCV device used for data protection must be fully allocated with the `allocate_type=persistent` option.
- A thin BCV device cannot be used while expanding a standard striped metadevice.
- Use of thin SRDF BCVs for data protection is not allowed.

The `protect_data` option is not applicable to unbound thin devices.

A thin BCV device used for data protection must be fully allocated with the persistent option specified (explained in "Creating thin devices (Enginuity 5773 through 5876)" on page 279).

As a part of the expansion process if the original metadevice was bound to a thin pool, the new metamember will be bound to the same pool.

Binding a thin metadevice to a pool

When binding a thin metadevice to a pool, use the `symconfigure` command to bind the metahead and give one preallocated size. Solutions Enabler will distribute the preallocated size among all the metamembers.

The distribution will depend on the type of the metadevice, as follows:

Striped metadevice — The preallocated size will be distributed evenly among the members.

Concatenated metadevice — Starting from the metahead, each member receives a minimum of the total preallocated size remaining, based on the specified `size_of_member`. For example, if 175 GB were preallocated for a metadevice with 3 members, each of 100 GB, Solutions Enabler will allocate 100 GB to the head and 75 GB to the 2nd member and 0 GB to the last member.

Refer to Chapter 8, "Configuring Virtual Provisioning" for more information about binding thin devices.

Converting devices

In the command file, you can convert the configuration type of an existing device or devices using the following command:

```
convert dev <SymDevName>[:<SymDevName>] to
          <DevConfig>
          [emulation=EmulationType,]
          [ ra_group=<n>, remote_dev=<SymDevName>,
            invalidate=<invalidate_opt>,
            [remote_mvs_ssid=<n>],
            start_copy=<YES | NO> ]
          [mvs_ssid=<n>] [raidset = [TRUE | FALSE]];
```

Where:

SymDevName — Specifies the device name of the device targeted for change. To target more than one device, indicate the first and last devices in a series separated by a colon (:).

DeviceConfig — Specifies the desired device configuration type.

For arrays running HYPERMAX OS 5997, Table 16 on page 238 lists allowable configuration conversions.

For arrays running Enginuity 5773 through 5876, Possible configuration conversions are shown in the first column of Table 11 on page 217.

`emulation` — Indicates the device's emulation type.

`ra_group` — Specifies the RA group number in the SRDF environment.

`remote_dev` — Specifies the remote array device name of the particular device targeted for change. If you specify a range of *SymDevNames* in the first line of the convert statement, the remote *SymDevName* value will be increased incrementally to arrive at the corresponding device number.

`invalidate_opt` — Indicates the SRDF device to invalidate so that a full copy can be initiated from the remote mirror. Allowed values are R1 (invalidate the source), or R2 (invalidate the target). The value NONE is not supported in Solutions Enabler V7.0 and higher.

`remote_mvs_ssid` — Specifies the remote z/OS (MVS) subsystem ID that will be assigned to any device created as a result of removing any mirror(s). If not provided, the original MVS SSID will be assigned when available. If the MVS SSID group is full, you must supply a new MVS SSID. Only one `remote_mvs_ssid` can be used in a session. It will be applied to all devices created within that session.

`start_copy` — Indicates whether an SRDF pair should be synchronized after the configuration change is committed.

Note: When creating SRDF devices, all conversions within a session must have:
- Device configuration settings that reflect the same destination SRDF type (RDF1 or RDF2).
- The same `ra_group` number.
- The same `invalidate` option.
- The same `start_copy` option.

`mvs_ssid` — Specifies the z/OS (MVS) subsystem ID that will be assigned to any device created as a result of removing any mirror(s). If not provided, the original MVS SSID will be assigned when available. If the MVS SSID group is full, you must supply a new MVS SSID. Only one `mvs_ssid` can be used in a session. It will be applied to all devices created within that session.

`raidset` — When requesting to convert a RAID-S group to unprotected devices, set `raidset` equal to `TRUE` and list the first RAID-S member. It is not necessary to list the other members.

To convert two existing BCV devices (0001C and 0001D) to an RDF1-BCV configuration and to invalidate the source R1 and synchronize the SRDF pair, enter in the command file:

```
convert dev 0001C:0001D to RDF1-BCV, ra group=1, remote_dev=0001c,
invalidate=R1, start_copy=YES;
```

An IBM i thin device (TDEV) can be converted to a BCV+TDEV device, and a BCV+TDEV device can be converted to a TDEV device.

Restrictions

The following are restrictions for converting devices:

- The `convert` command can be used for three different classes of device configuration changes, as long as the class types are performed in separate sessions.

  Note: Changes for multiple operation classes can be executed in the same session, except for dynamic SRDF changes and device pool changes.

  The three class types that cannot be used in the same session are:

  - Add/remove BCV/DRV attributes
  - Add/remove SRDF attributes
  - Increase/decrease mirroring

- Full swap operations require the R1 and R2 devices to be the same size.
- You cannot convert one member of a raidset to unprotected without converting all the members to unprotected.
- When adding/removing SRDF attributes, there are no restrictions on I/O. The SRDF pair must be split or failed over. If failed over, the R1 device must be unmapped.
- When adding/removing BCV attributes, there are no restrictions on I/O. The standard/BCV pair must be split.
- The SRDF mode on an SRDF device pairing will be Adaptive Copy Disk by default, unless the SYMAPI_DEFAULT_RDF_MODE is set in the option file. If the device being converted is a diskless R1 device, the RDF mode will default to Adaptive Copy Write Pending, regardless of the option file setting. (Applies to arrays running Enginuity 5876.)
- For arrays running Enginuity 5773 through 5876:
  - When adding DRV attributes, devices must be unmapped.
  - Only the head of a metadevice can have its type changed. The metamember(s) will automatically have the changes applied.

Converting thin devices

Thin devices can be converted to other thin device configurations. Table 16 lists the valid conversions.

Table 16 Thin device conversions

Original thin device[a]	Converted to
TDEV[b]	RDF1+TDEV
TDEV[b]	RDF2+TDEV
TDEV	BCV+TDEV

Table 16 Thin device conversions

Original thin device[a]	Converted to
BCV+TDEV[c]	TDEV
RDF1+TDEV[b]	TDEV
RDF2+TDEV[b]	TDEV
BCV+TDEV[b]	RDF1-BCV+TDEV
BCV+TDEV[b]	RDF2-BCV+TDEV
RDF1+TDEV[b]	RDF1-BCV+TDEV
RDF2+TDEV[b]	RDF2-BCV+TDEV
R1-BCV+TDEV[bd]	R1+TDEV
R2-BCV+TDEV[bd]	R2+TDEV
R1+TDEV[d]	R1-BCV+TDEV
R2+TDEV[d]	R2-BCV+TDEV

a. The original device type should be created before converting the device.

b. Not for VMAX 10K Series systems; all devices are dynamic RDF capable by default.

c. BCV+TDEV to TDEV is the only supported conversion for a BCV+TDEV device.

d. Requires Enginuity 5875 Q32011 SR.

For example, to convert a thin device TDEV to an RDF1 thin device:

```
convert dev 00015 to RDF1+TDEV;
```

For example, to create a thin R1 BCV device by converting a thin BCV device:

```
symconfigure -sid 397 -nop -v -cmd "convert dev 015F2 to RDF1-BCV+TDEV ra_group=10
   remote_dev=16A invalidate=R1 start_copy=no;" commit
```

Impact on I/O

The following are restrictions/conditions to avoid impact on I/O activity:

- The BCV attribute is not allowed on thin SRDF devices or thin dynamic SRDF devices.
- The device being converted must not be part of a clone session.

Device group invalid warning (Enginuity 5773 through 5876)

When converting devices that are currently in a device group, some groups will be declared invalid after particular operations.

One example would be if you are performing TimeFinder operations and have placed in a group, one standard (STD) device and one TimeFinder (BCV) device. When `symconfigure` enables you to change the BCV device to a non BCV device, the conversion leaves the group in an invalid state.

There are two ways to handle this situation:

- Remove the TimeFinder (BCV) device from the group before you begin the conversion process.

- Or, after the device has been converted, export (`symdg export`) the device group and remove from the file `symdg.txt` the invalid device and import (`symdg import`) the device group.

Converting a metadevice (Enginuity 5773 through 5876)

To convert a metadevice's configuration from concatenated to striped, use the following command:

```
convert meta SymDevName config=MetaOption
[, stripe_size=MetaStripeSize>[cyl]],
[, protect_data=[TRUE|FALSE],
[bcv_meta_head=SymDevName];
```

Where:

`stripe_size` — Specifies the stripe size in 512 byte blocks.

`protect_data` — Specifies whether the data needs to be protected. Possible values are `TRUE` or `FALSE`. This option can only be used while converting a metadevice from concatenated to striped.

`bcv_meta_head` — Specifies the device name (or thin device name).

Metadevices can be converted from concatenated metadevices to striped devices using the `convert meta` command. The command can also be used to change the stripe size of a striped metadevice.

When converting from concatenated metadevices to striped devices, use the `protect_data` option to preserve data on the BCV metadevice during conversion. The BCV metadevice must match the original standard meta in size, configuration, and member count.

The `bcv_meta_head` option specifies which BCV metadevice to use.

To convert a concatenated metadevice 00030 to a striped metadevice and preserve data during the conversion using BCV metadevice 01F, enter:

```
convert meta 00030, config=striped, stripe_size=1920,
protect_data=TRUE, bcv_meta_head=01f
```

Note: The specified BCV metadevice must be identical to the original metadevice in capacity, stripe count, and stripe size.

Thin metadevices (Enginuity 5773 through 5876)

The following thin metadevice conversions are supported:

- Convert thin concatenated metadevices to thin striped metadevices, without protecting data.
- Convert thin concatenated metadevices to thin striped metadevices, while protecting data using a regular BCV device.
- Convert a thin concatenated metadevice to a thin striped metadevice, while protecting the data on the thin meta using a thin BCV device that is persistently allocated.
- Convert a thin striped metadevice to a thin concatenated device, without protecting the data.

Setting device attributes

You can set the device attributes or emulation of a number of devices in a range using the following command:

```
set device SymDevName[:SymDevName]
      [emulation=EmulationType]
      [identity = NO identity]
      [attribute=[NO] device_attr];
```

Where:

`emulation` — Specifies the device emulation type, which can be the following: `FBA` or `CELERRA_FBA`.

Note: You cannot set the device emulation type for Data Domain devices.

`identity` — Restores the devices identity to its original value.

`attribute` — Indicates if a device attribute restricts how a device can be accessed.

For arrays running HYPERMAX OS 5977, Possible values include:

- SCSI3_persist_reserv
- DIF1
- AS400_GK

Note: You cannot set device attributes for Data Domain devices.

For arrays running Enginuity 5773 through 5876, possible values include:

- `RDB_Cksum` (Enginuity 5773 and lower)
- `VCMdb` (for device masking on Enginuity 5773 and lower)
- `SCSI3_persist_reserv` (persistent group reservation) In Enginuity 5874 and lower, this attribute is disabled by default. In Enginuity 5875 and 5876, this attribute is enabled by default. With Enginuity 5876 Q42012SR, you cannot set or clear the SCSI-3 persistence attribute for mapped devices; you must unmap the device prior to setting or clearing this attribute.
- `dyn_rdf` (This option provides the most flexibility in performing dynamic SRDF operations)
- `dyn_rdf1_only`
- `dyn_rdf2_only`
- `ACLX` (for Auto-Provisioning Groups on Enginuity 5874 and higher)
- `RCVRPNT_TAG` (for RecoverPoint on Enginuity 5875.198.148 and higher)
- `DIF1` (Enginuity 5876)
- `AS400_GK` (Enginuity 5876)

Note: You can specify the `dyn_rdf`, `dyn_rdf1_only`, and `dyn_rdf2_only` attributes for thin BCV devices.

To convert five devices (00015 to 00019) to Celerra FBA emulation, enter in the command file:

```
set device 00015:00019 emulation=CELERRA_FBA;
```

Restrictions

The following restrictions apply when setting device attributes:

- A device that is mapped or masked to an FCoE port cannot have the RCVRPVT_TAG attribute.
- When setting the device emulation type, the devices must be unmapped. No I/O to the devices involved.
- When setting the attribute type to a mapped device, it is recommended that you minimize the I/O activity to the affected devices.

The following restrictions apply to setting the DIF1 device attribute:

- The DIF1 attribute can only be set on FBA devices.
- DIF1 attribute can be set on both standard and thin host-accessible devices. You cannot set the DIF1 attribute on any internal devices.
- A device must be unmapped if resetting the DIF1 attribute.
- A device with the DIF1 attribute can only be mapped to fiber front-end directors (no iSCSI or FCoE).
- Metadevices with the DIF1 attribute must have the same state, either all set or all reset, on the metahead and all metamembers.
- DIF1 attribute can not be set on DATA and SAVE devices.
- Devices can have either the RDB_Checksum attribute or the DIF1 attribute, not both. The DIF1 flag cannot be set on a device with an active double checksum.
- Devices can have either ACLX attribute or the DIF1 attribute, not both.
- There is no relation between the DIF1 attribute and replication. Both source and target devices of any replication can have their own DIF1 setting.
- The DIF1 attribute needs to be set before requesting a reset. If the reset request is for a device range, and any one of the devices does not have the DIF1 attribute set, an error returns.

The following restrictions apply to setting the AS400_GK attribute:

- The AS400_GK attribute is only supported with Enginuity 5876 and higher.
- The AS400_GK attribute cannot be set on a D910_099 thin device unless the thin device is bound to a pool.
- The AS400_GK attribute can only be set on AS/400 or Celerra FBA devices.
- The device can not be configured as SAVE, DATA, BCV, SRDF or dynamic RDF.
- The device can be member of an SG, DG, or CG.
- If an AS/400 or a Celerra FBA device is created using the command `create gatekeeper`, the AS400_GK attribute is automatically enabled.
- The AS400_GK attribute can only be set on non metadevices. This attribute is only supported on AS400 devices with the following emulation types:
  - AS/400_M2107_A02
  - AS/400_M2107_A04

- AS/400_M2107_A05
- AS/400_M2107_A06
- AS/400_M2107_A07
- AS/400_M2107_A82
- AS/400_M2107_A84
- AS/400_M2107_A85
- AS/400_M2107_A86
- AS/400_M2107_A87
- AS/400_M2107_050
- AS/400_M2107_099
- AS/400_D910_099

Setting device geometry (Enginuity 5773 through 5876)

You can set the geometry for a disk on arrays managed by a Solaris host.

Note: This feature is only for Solaris hosts using the `format` utility.

Arrays running Enginuity 5773 will display (with `list` and `show` commands) the specified (emulated) device geometry. Therefore, when a device's geometry is changed to emulate an a different device type, the display does not show the native, or true geometry, but the one that was specified.

Note: Throughout this section the following terms will be used:
`Symm-6` will be used to represent arrays with Enginuity versions 56*xx* and earlier, and `Symm-7` will represent arrays with Enginuity version 57*xx* through 5876.

Some factors to consider when using the disk geometry feature are:

- The disk geometry specified will remain as the geometry for the disk, until it is explicitly changed again.
- If a copy of a target disk is made, you need to remember to change the geometry of the new target too. As an example, if a device is replicated using SRDF from `Symm-6` (source) to `Symm-7`, you may decide to change the geometry on the replicated `Symm-7` device. If the `Symm-7` device were to be cloned, the clone device does not automatically receive the emulated geometry—it must be explicitly set.
- You need to know the exact geometry of the source disk and need to ensure that before the migration or replication occurs, this geometry is set for the target disk.

Note: There is also an array-wide setting called `FBA Geometry Emulation` that can only be turned on by an EMC Customer Engineer. When this is set to `Enabled`, all devices on a Symm-7 will emulate Symm-6 geometry. When set to `Enabled` on an array running Enginuity 5874, all devices will emulate Symm-6 geometry. However, even if this setting is enabled, the option of setting a geometry on an individual device level still exists, and this setting will take precedence over the array-wide setting.

This feature is set using the `symconfigure` command, which is executed using a command file. The syntax for setting the device geometry follows:

```
set dev SymDevName[:SymDevName]
   geometry [= SYMM-6 | = SYMM-7 | = CLARIION]
   cyls = n;

set dev SymDevName[:SymDevName]
   geometry = <NO GEOMETRY>;
```

Where:

`SYMM-6` — Refers to the device geometry in arrays running Enginuity 56xx and earlier (64 sectors/track, 15 tracks/cylinder).

`SYMM-7` — Refers to the device geometry in arrays running Enginuity 57*xx* and higher (128 sectors/track, 15 tracks/cylinder).

`CLARIION` — Refers to the CLARiiON disk geometry (10 sectors/track, 16 tracks/cylinder; each sector is 512 bytes).

n — Refers to the device cylinder size.

`NO GEOMETRY` — Clears the current geometry if the device has an emulated geometry set.

Blocks and sector size for an FBA device is 512 bytes. The device geometry cannot be set on the following devices:

- Metamember
- Vault device
- SFS device
- SAVE device
- DATA device
- DRV
- Mapped device
- Non-FBA device
- VCMDB device

Showing device geometry

Disk geometry parameters will display if the `-v` option or the `-geometry` option is specified for the following commands:

- `symdg list -v`
- `symdev list -v`
- `sympd list -v`
- `symdev show -geometry`
- `sympd show -geometry`

The following is an example of the command for displaying the device geometry:

```
symdev -sid 016 show 0317 -geometry

Device Physical Name     : Not Visible

 Device Symmetrix Name    : 00317
 Device Serial ID         : N/A
 Symmetrix ID             : 000192600175

 Number of RAID Groups    : 1
 Attached BCV Device      : N/A
```

```
 Attached VDEV TGT Device : N/A

 Vendor ID                : EMC
 Product ID               : SYMMETRIX
 Product Revision         : 5874
 Device WWN               : 60000970000192600175533030333137
 Device Emulation Type    : FBA
 Device Defined Label Type: N/A
 Device Defined Label     : N/A
 Device Sub System Id     : 0x1000
 Cache Partition Name     : DEFAULT_PARTITION

 Device Block Size        : 512

 Device Capacity
     {
     Cylinders            :        958
     Tracks               :      14370
     512-byte Blocks      :    1839360
     MegaBytes            :        898
     KiloBytes            :     919680
     }

 Effective Device Geometry: Native
     {
     Sectors/Track        :        128
     Tracks/Cylinder      :         15
     Cylinders            :        958
     512-byte Blocks      :    1839360
     MegaBytes            :        898
     KiloBytes            :     919680
     }

…(output shortened for this example...)
```

The field Effective Device Geometry can have four possible values:

- `User Defined` — Indicates the user has defined a geometry for this device.
- `Native` — Indicates the current device geometry is the same as the native geometry for the array.
- `Array wide emulation` — Indicates that the array-wide flag for FBA Geometry Emulation is set to Enabled and the effective geometry shown in the output is derived from this setting.

  Note: Even if the array-wide setting is turned on, the user can still define the geometry at the individual device level and this setting will take precedence over the array-wide setting.

- `N/A` — For non FBA devices.

Setting device identifiers

Solutions Enabler supports device identifier management on arrays. This support allows you to define names and identifiers for EMC array devices, HP devices, and VMS devices. The definitions are configured in a command file and processed with the `symconfigure` command.

The format of the command file for setting device identifiers follows:

```
set dev SymDevName[:SymDevName]
[[device_name = 'DevName'] | [NO DevName]]
[[hp_identifier = 'hp_id'] | [NO hp_identifier]]
[[vms_identifier = vms_id] | [NO vms_identifier]];
```

This format accommodates setting a device name or identifier on a single device, a range of devices, or multiple ranges of devices. Device identifiers do not have to be unique for all devices.

Restrictions for device_name

- The `device_name` can be less than or equal to 64 characters in length.
- Any character may be used except quotes, which are used to mark the start and end of the input.
- The names are case sensitive.
- There is no default device name for a device.
- The `device_name` parameter is supported only for Enginuity 5874 or higher.

Restrictions for hp_id

- The `hp_id` can be less than or equal to 128 characters in length.
- Any character may be used except quotes which are used to mark the start and end of the input.
- The identifiers are case sensitive.
- There is no default HP identifier for a device.

Restrictions for vms_id

- The `vms_id` here is an unsigned 4-byte integer. A `vms_id` can be a number between 0 and 32766. This value will be accepted and displayed as a decimal value.

Note: The device identifier can include a `device_name` and the `hp_id` or a device name and the `vms_id`. The device identifier cannot include both the `hp_id` and the `vms_id`.

Identifier exclusions

You cannot set device identifiers on the following devices:

- Metamembers. The device identifier of the metahead device will display, if applicable.
- The following internal devices (device identifiers display as N/A):
  - VAULT
  - SFS
  - DRV
  - SAVE (device names will be allowed for this device type)
  - DATA (device names will be allowed for this device type)

Viewing device identifiers

Device identifiers are viewed through the `symdev list` command. CLI commands are usually limited to an 80-character output, but if the user requests the `hp_id` of a device to be displayed, the line could be over 80 characters in length. A new form of the `symdev list` command displays the device identifiers.

This CLI command does not work in offline mode.

Note: Device name, device nice name, HP device identifiers, and VMS device identifiers cannot be used in any control command. All these device identifiers are only for display. Both HP device identifiers and VMS device identifiers can be set on any host and any devices.

Deleting devices

In the command file, you can delete one or more devices from the specified array. Deleting a device frees the space that it once occupied for future use.

To delete one or more devices, use the following command:

```
delete dev SymDevName[:SymDevName]
```

For example, to delete device `00015` from array `345`, create a command file containing the following:

```
delete dev 00015;
```

Then activate the option using the command:

```
symconfigure -sid 345 -file delete_dev.cmd -v -noprompt commit
```

Enginuity versions 5773 through 5876

The raidset option is available with arrays running Enginuity 5773 through 5876. To delete one or more devices, use the following command:

```
delete dev SymDevName[:SymDevName][,raidset = TRUE | FALSE];
```

When requesting to delete a RAID-S group, set `raidset` to `TRUE` and list the first RAID-S member. It is not necessary to list the other members; all will be deleted.

For example, to delete all devices in a RAID-S set, create a command file with a statement that only identifies the first member of the RAID-S set:

```
delete dev 0015C, raidset = TRUE;
```

Then run the `symconfigure` utility:

```
symconfigure -sid 605 -file delete_raids.cmd -v -noprompt commit
```

All members will be deleted.

Restrictions (HYPERMAX OS 5977)

The `delete dev` command is blocked if the source device specified by *`SymDevName`* is a TDAT device.

The device must not be mapped to a front-end port.

Restrictions (Enginuity 5773 through 5876)

The following are restrictions for deleting devices with Enginuity 5773 through 5876:

- The device emulation must be one of the following types:
  - FBA
  - CELERRA_FBA
  - VME_512_FBA
  - CKD_3380
  - CKD_3390
  - AS400_*n*

  Note: AS/400 metadevices must be dissolved before being deleted.

- The device must not have an attached BCV or DRV.
- The device must not have any snap sessions.
- If the device is a DATA device, it must be disabled and have no used tracks.
- The device must not be:
  - Mapped to a front-end port
  - Part of an SRDF consistency group
  - A source or target of a clone session
  - Held
  - A virtual device that is in use
  - An RDF, SFS, SAVE, or VCM database device
  - A metamember
  - Masked by VCM
  - Bound if a thin device
  - Part of a storage group or view

Managing device reservations

You can use the configuration change functionality to reserve devices and front-end mapping addresses for future configuration and masking operations. When using this feature, you reserve the devices/addresses you plan on using, verify that no one else has reserved the resources, and release the reservations when the task is complete.

All reservations are assigned a reserve ID, indicating that the specified devices/addresses are reserved. Any attempt to use the reserved devices/addresses will return a message indicating that the devices/addresses are reserved.

There are two types of device reservations:

- **Enforced** — Reservations are enforced by the SYMAPI library, and require that you specify the reserve ID to use the devices. This is the default behavior when reserving devices.
- **Advisory** — Reservations are enforced by co-operating applications. Some applications can ignore advisory reservations, allowing knowledgeable users to make configuration changes on reserved devices, provided that their changes are compatible with the reserving task's goal.

Both types of reservations can have expiration dates associated with them, which will automatically release a reservation if you fail to explicitly do so.

Device reservations are honored only when devices are explicitly specified during Solutions Enabler configuration change operations. Operations that allow the Enginuity operating environment to choose devices (such as when a meta is formed and only the metahead is specified) do not honor device reservations.

Device reservations are enabled (TRUE) by default in the `options` file. To disable device reservations, set the SYMAPI_ENABLE_DEVICE_RESERVATIONS parameter in the `options` file to FALSE.

Device reservations are enforced (TRUE) by default in the `options` file. To disable the enforcing of device reservations, set the SYMAPI_ENFORCE_DEVICE_ RESERVATIONS parameter in the `options` file to FALSE.

Note: For information on changing the `options` file parameters, refer to the *EMC Solutions Enabler Installation Guide* or the *EMC Solutions Enabler SYMCLI Command Reference* .

Reserving devices

To reserve devices, use the following form:

```
symconfigure -sid SymmID [-expire ExpirationDate]
             [-f[ile] CmdFile | 'redirect stdin' | -cmd "Cmd"]
            -owner Owner -comment UserComment
            [-enforce | -advise]
    reserve
```

Where:

`ExpirationDate` — Specifies the date and time for a device reservation to expire. This is an optional parameter, and if not specified, defaults to no expiration. The format for this parameter is:

[*mm*/*dd*[/*yy*]:][*hh*:*mm*[:*ss*]]

If you only provide the *`hh:mm`*, the current day will be assumed. If you only provide the *`mm/dd`*, the current year will be assumed. You can also specify a four-digit year.

`CmdFile` — Specifies the name of any ASCII text file containing a set of commands to process at a higher time. This file can be used in the following ways:

- To reserve devices for specific configuration change operations, in which case the file will list configuration change commands.
- To reserve devices for non-specific operations, in which case the contents of the file will use the following form:

  ```
  reserve dev SymDevName[:SymDevName]
  ```

  Using this method allows you to reserve devices for other applications.

  Note: For arrays running HYPERMAX OS 5977, the `reserve dev` operation is blocked if the source device specified by *`SymDevName`* is a TDAT device.

Owner — Specifies the name of the owner of the reservation (up to 31 characters long).

Usercomment — Indicates a user-specified comment detailing the device reservation (up to 255 characters long).

-enforce — Specifies an enforced reservation (default).

-advise — Specifies an advisory reservation.

When reserving devices, be sure to note the returned reserve ID for subsequent processing.

Committing changes on reserved devices

When committing changes on devices reserved with the Enforced flag, you must supply the appropriate reserve ID. If you do not have the reserve ID and someone else has reserved the devices, the commit will fail. If you have reserved the devices, or no one else has reserved the devices, then the commit will succeed.

When committing changes on devices reserved with the Advisory flag, some applications may not require a reserve ID. However, the symconfigure command does require a reserve ID.

To commit changes on devices reserved with the -enforce option, use the following form:

```
symconfigure -sid SymmID -f[ile] CmdFile commit
[-reserve_id=ResvID[,ResvID[,ResvID]]]
[-remote_reserve_id=ResvID[,ResvID[,ResvID]]]
```

For example, to commit the changes in the command file delete.cmd, enter:

```
symconfigure -sid 3241 -file delete.cmd commit -reserve_id 5
```

Viewing existing reserved devices

To view the devices reserved on an array, use the following command:

```
symconfigure -sid SymmID -reserved list
```

To view details on a specific reservation ID, use the following command:

```
symconfigure -reserve_id ResvID show
```

Where:

ResvID — The device reservation ID.

Releasing reserved devices

When releasing device reservations, you must supply the appropriate reserve ID. Performing a configuration change on reserved devices will not release them. You must do this as an independent step.

To release reserved devices, use the following form:

```
symconfigure -sid SymmID [-noprompt] -reserve_id
ResvID[,ResvID[,ResvID]]
    release
```

Where:

ResvID — Specifies the device reservation ID.

For example, to release the set of devices with the ID 5 and 7, enter:

```
symconfigure -sid 3241 release -reserve_id 5,7
```

Managing device pools (HYPERMAX OS 5977)

Arrays running HYPERMAX OS 5977 are shipped with pre-configured thin pools. During the pre-configuration process, the physical disks are partitioned into a set of internal disk groups based on the technology, capacity, and rotational speed of the disk and protection scheme desired. In addition each disk group is fully provisioned with DATA devices, thin pools are created, and the data devices are assigned to the thin pool created for that disk group.

If physical disks are added after system delivery, the install process either creates one or more new disk groups, DATA devices, and thin pools to accommodate those disks, or adds the physical disks to existing disk pools, provisions them into DATA devices consistent with the disk group definition and adds the devices to the pool for that disk group.

For details on HYPERMAX OS 5977 thin pool provisioning and restrictions, refer to "Overview for arrays running HYPERMAX OS 5977" on page 276.

Managing device pools (Enginuity 5773 through 5876)

There are three types of device pools used for managing data; Snap pools, SRDF/A DSE (Delta Set Extension) pools, and thin pools.

- Snap pools, used for TimeFinder/Snap operations, store pre-update images or change tracks during a virtual copy session. The array supports the creation of multiple named device pools, allowing `symsnap` commands to select a particular pool. This alleviates contention for SAVE device space among several snap sessions and eliminates the possibility of a single session using all pool space.

  For more information about SAVE devices and TimeFinder/Snap operations, refer to the *EMC Solutions Enabler TimeFinder Family CLI Product Guide*.

- SRDF/A DSE pools used for SRDF operations extends the cache space available to SRDF/A cycles by off-loading some or all of the cycle data from cache to pre-configured disk storage pool. These pools are configured the same as Snap pools.

  For information about using SRDF/A DSE pools, refer to the *EMC Solutions Enabler SRDF Family CLI Product Guide*.

- Thin pools used for Virtual Provisioning operations contain DATA devices that are bound to thin devices. Each write to a thin device by the host is written in the bound DATA device pool. The DATA device pools do not have a default pool.

  For information on Virtual Provisioning and thin pools, refer to "Configuring Virtual Provisioning" on page 275.

Device pool attributes and rules

All device pools

The following attributes and rules apply to all device pool types:

- All device pools must have the same disk type (EFD, FC, SATA), unless the SYMAPI_POOL_ALLOW_MIX_TYPE options file setting is ENABLE.
- A device pool can only have devices of one emulation type in them. The pool itself is not defined to be a pool for an emulation type. Instead, the first device added to the pool defines its emulation type.
- When a pool is created, it will be created as either: `snap`, `rdfa_dse`, or `thin` pool.
- Device pools are made up of devices that are specifically configured to be used for pools.
- Devices that do not belong to any pools are available to be added to a pool. Devices in a pool are available for use once they are enabled.
- Devices can be removed from a pool, but must be disabled and drained.

Snap pools

The following attributes and rules apply to snap pool types:

- A special Snap pool DEFAULT_POOL exists by default, which contains SAVE devices that have not been assigned to any named pools but are available for Snap operations.
- A disk group must be specified when creating a SAVE device if the `create dev` command is also adding the device(s) to a pool.

SRDF/DSE pools

The following attributes apply to SRDF/A DSE pool types:

- SRDF/A DSE pools can be associated (shared) with multiple SRDF/A groups.
- A single SRDF/A group can have, at most, four RDFA DSE pools associated with it. There can be at most one FBA, one CKD3390, one CKD3380, and one AS400 pool.
- An SRDF/A DSE pool can contain only one type of device from the list of FBA, CKD3390, CKD3380, and AS400.

Creating a device pool

To create a snap or RDF/A DSE device pool, use the following form:

```
create pool PoolName, type = snap | rdfa_dse
```

The maximum number of device pools per array is 512.

For example, to create a new pool named `Finance`, enter:

```
create pool Finance, type=snap;
```

Deleting a device pool

You can only delete a device pool if it does not have any members.

To delete a device pool, use the following form:

```
delete pool PoolName, type= snap | rdfa_dse | thin;
```

For example, to delete a pool named `Finance`, enter:

```
delete pool Finance, type=snap;
```

Adding devices to a pool

You can only add disabled and inactive devices to a pool. While adding the devices, you can specify whether they should be enabled or disabled in the pool.

To add devices to a pool, use the following form:

```
add dev SymDevName[:SymDevName] to pool PoolName
type= snap | rdfa_dse | thin
[, member_state=ENABLE | DISABLE];
```

Where:

`member_state` — Indicates whether the device(s) being added should be enabled or disabled in the pool.

To add and enable devices from `0001B` to `0001D` to a thin provisioning data pool named `ACT20`, enter:

```
add dev 0001B:0001D to pool ACT20, type=thin, member_state=ENABLE;
```

Removing devices from a pool

You can only remove disabled and inactive devices from a pool. Once you remove a device from a pool, the device is available to be added to any other pool.

To remove devices from a pool, use the following form:

```
remove dev SymDevName[:SymDevName] from pool PoolName,
type=snap | rdfa_dse | thin;
```

To remove device `0001E` from a pool named `HR30`, enter:

```
remove dev 0001E from pool HR30, type=rdfa_dse;
```

Note: The last DATA device of a pool in a thin tier cannot be removed, even if that pool has no bound devices.

Enabling/disabling devices

You can enable or disable devices for use in a pool. The devices in a pool do not all have to be in the same state (enabled or disabled). If all the devices in a pool are disabled, then the pool is disabled. If at least one device in a pool is enabled, then the pool is enabled. To enable or disable a range of devices, all the devices must be in the same pool. To disable a device, all sessions must be terminated, and have no used tracks.

Note: The last DATA device of a pool in a thin tier cannot be disabled, even if that pool has no bound devices.

If a DATA or SAVE device has used tracks and is disabled, the data will be drained to other devices in the pool if there is enough space. When complete, the device state changes to Disabled.

To enable or disable a device in a pool, use the following form:

```
enable | disable dev SymDevName[:SymDevName] in pool PoolName,
type=snap | rdfa_dse | thin;
```

Where:

PoolName — Specifies the name of the device pool. It can be from 1 to 12 alphanumeric characters long and include hyphen (-), and underscore (_) characters.

For example, to enable the use of three new DATA devices in a pool named `Finance`, enter:

```
enable dev 0001B:0001D in pool Finance, type=thin;
```

Renaming a device pool

To rename a device pool, use the following form:

```
rename pool PoolName to NewPoolName type = <snap | rdfa_dse | thin>;
```

Restrictions

- The array must be running Enginuity 5875 or 5876.
- The new pool name must adhere to the same naming restrictions as when creating a pool.
- In a single command file that includes operations on a pool and a pool rename, you can:
  - Use the old pool name for the pool operations and rename the pool as the last operation.
  - Rename the pool as the first operation and use the new name for the subsequent pool operations.

  You cannot use the old pool name for some operations, change the pool name, and then use the new pool name for subsequent operations.
- Snap, thin, and DSE pools can be renamed; however, the default pool name (DEFAULT_POOL) for snap cannot be changed.
- You cannot create and rename a pool in the same session.

Monitoring a device pool

To monitor a device pool's usage, use the `symcfg monitor` command:

```
symcfg [-sid SymmID] [-i Interval] [-c Count] [-offline]
         [-percent <1-100> -action ScriptFile [-norepeat]]
         [-snap | -rdfa_dse | -thin] [-pool PoolName]
         [-mb] [-gb]
       monitor
```

Where:

`-action` — Selects a script that should be run when the specified percent value is encountered. The full pathname to the action script must be specified. The first argument passed to the script is automatically set to the percent value.

`-c` — Indicates the number (count) of times to display or to acquire an exclusive lock on the array host database, the local array, and the remote arrays.

`-i` — Specifies the repeat interval in seconds to display or to acquire an exclusive lock on the array host database, the local array, and the remote arrays. The default interval is 10 seconds. The minimum interval is 5 seconds.

`monitor` — Checks the total percent full of the devices currently configured in the pool and can optionally execute a script file if a specified percentage is encountered.

`-norepeat` — Specifies that the action script should only be run once if the threshold has been met. Used with the action script option on the `monitor` command.

`-percent`— Causes the action script to be executed when the percent full argument is encountered.

`-pool` — Specifies an SRDF/A DSE, snap, or thin pool.

`-snap` — Specifies that the pool type is a snap pool.

`-rdfa_dse` — Specifies that the pool type is an RDF/A DSE pool.

`-thin` — Specifies that the pool type is a thin pool.

Viewing a device pool

For example, to display all pools, use the `symcfg list` command with the array ID:

```
symcfg list -sid 397 -pools

Symmetrix ID: 000194900397

                        S Y M M E T R I X   P O O L S
--------------------------------------------------------------------------------
             T T                                                        F   S
             y e                                                        u   t
 Pool        p c Dev   Dev             Total  Enabled     Used     Free ll  a
 Name        e h Emul  Config         Tracks   Tracks   Tracks   Tracks (%) te
------------ - - ----- ------------ -------- -------- -------- -------- --- ---
DEFAULT_POOL S M FBA   2-Way Mir       50775    50775        0    50775   0 Ena
  . . .
CKD_POOL     T S 3390  2-Way Mir       29376    29376      240    29136   0 Ena
TR1_POOL     T S FBA   2-Way Mir     3486576  2586576   115260  2471316   4 Ena
  . . .
testPRC      T - N/A   Unknown             0        0        0        0   0 Dis
test_prc     T S FBA   2-Way Mir       14700        0        0        0   0 Dis
  . . .
HR_POOL      T S 3390  2-Way Mir      146880   146880      960   145920   0 Ena
  . . .

Total                                -------- -------- -------- -------- ---
Tracks                                3957531  3018099   116820  2901279   3

Legend:
  Pool (Type):
    S = Snap,  R = Rdfa DSE  T = Thin
  (Tech)nology:
    S = SATA, F = Fibre Channel, E = Enterprise Flash Drive, M = Mixed, - = N/A
```

Some notes on the example output follow:

Note that the pool state is Disabled, if all devices in the pool are disabled. The pool state is enabled, if any device is enabled.

Free Tracks is (Enabled - Used) the disabled tracks are not counted when counting free tracks.

Full % is (Used Tracks/Enabled Tracks * 100) disabled tracks are not counted.

The Technology type is new with Enginuity 5875.

Mapping devices

You can map devices to front-end director ports, or a range of devices to consecutive addresses specifying a starting address. This is part of the Device Reallocation feature.

To map a device to a director port, use the following command:

```
map dev SymDevName [:SymDevName]to dir DirectorNum:PortNum
[, emulation=EmulationType]
[starting][target=ScsiTarget,] lun=ScsiLun
[, vbus=FibreVbus]
```

You can enter the parameters after the `map dev` command in any order.

Where:

`emulation` — Indicates the device's emulation type. This option is required when performing operations on a Celerra device, and indicates that you are aware that you are changing the Celerra environment. Solutions Enabler supports mapping IBM i thin devices.

`lun` — Specifies the SCSI logical unit number (hex value).

`starting` — Specifies the starting address for the range of devices.

`target` — Specifies the SCSI target ID (hex value).

`vbus` — Specifies the virtual bus (vbus) address for mapping to an FA port if using volume set addressing.

For example, to map device `00030` to director `16A`, port 0, and SCSI target/lun 0, 7, enter in the command file:

```
map dev 00030 to dir 16A:0 target=0, lun=7;
```

To obtain a list of used addresses, including the next available address, use the following command:

```
symcfg list -SA all -address -available
```

To map a device `00032` to director 16A, port 0, and SCSI target/lun 0,2 and update the device masking database by specifying the WWN `20000000c920b484` of the host bus adapter (HBA) port through which a host accesses the device, enter the following command:

```
map dev 00032 to dir 16A:0 target=0, lun=2, wwn=20000000c920b484;
```

Restrictions

- When mapping, there are no restrictions on I/O if adding a second path.

- After committing a `symconfigure` mapping operation, you must update the device mapping information within the host system environment. Attempting host activity with a device after it has been removed or altered, but before you have updated the host's device information, can cause host errors.
- To update your hosts, run the utilities specified for your platform as described in "Introducing devices to a host" on page 261. After the host environment is updated, I/O activity can resume with the array device.
- For arrays running HYPERMAX OS 5977, the following apply:
  - The `map dev` command returns a feature not supported error when mapping devices to ACLX enabled front end ports.
  - The `map dev` command returns a feature not supported error if the devices are encapsulated devices being used as TimeFinder VX source devices.
  - The valid range for *`PortNum`* is extended to 0 to 31.
- For arrays running Enginuity 5875 or lower, mapping a RecoverPoint-tagged device or an AS/400 device to an FCoE director is only supported .

Mapping CKD devices (Enginuity 5773 through 5876)

You can map or unmap a range of CKD devices from an EA/EF port. In addition, you can also assign PAV alias addresses for devices already mapped.

Note: The mapping of CKD devices is not supported for arrays running HYPERMAX OS 5977.

Mapping a range of devices

To map a range of devices to an EA/EF port, use the following form:

```
map dev SymDevName[:SymDevName] to dir
DirectorNum:PortNum, starting
base_address=cuu_address [mvs_ssid=nnn];
```

To map a range of devices to an EA or EF port, using the same addressing from a different port, use the following form:

```
map dev SymDevName[:SymDevName] to dir
DirectorNum:PortNum, copying dir
DirectorNum:PortNum;
```

Where:

`base_address` — Specifies the base address to be assigned to the first device in the mapping request. It will be incremented by one for each device in the range of devices being mapped.

Mainframe ports expect devices to be mapped in groups to form CU images. The first digit in the address is the CU image number, which can range from 0 to FF (Enginuity 5875). The remaining two digits can range from 00 to 0xFF.

You can divide the set of 256 `cu_address` available in a CU image into base addresses and aliases, as is required.

`mvs_ssid` — When mapping a range of devices to an EA/EF port, you may need to change the current `mvs_ssid` assigned to the devices. If the devices are becoming part of an existing CU image, they will be assigned the `mvs_ssid` of the devices already mapped. If a new CU image is being formed and mapped, a new `mvs_ssid` can be assigned during the map request.

`copying dir` — Specifies a directory from which to copy; used with Enginuity version 5671 for each EA/EF port.

To map devices 0001D through 00023 to CU image A, where the next free base address is 20 and the SSID is 140, enter:

```
map dev 0001D:00023 to dir 01C:1 starting base_address=A20
mvs_ssid=140;
```

Copying the mapping of a range of devices

To copy a CU image mapping, all devices in the range must be mapped to the same CU image, or not mapped at all. Devices within the specified range that are not mapped will be ignored as long as they are not mappable (SAVE devices, DRVs, and so on). If a device in the range is not mappable, the request will be rejected.

To copy a CU image mapping from one port to another, use the following form:

```
map dev SymDevName[:SymDevName] to dir DirectorNum:PortNum
copying dir DirectorNum:PortNum
```

To copy addresses of devices `00010` through `00050` from port `03C:0` to port `03D:0`, enter:

```
map dev 00010:00050 to dir 03D:0 copying dir 03C:0;
```

Assigning PAV alias addresses to mapped devices

When assigning PAV alias addresses to mapped devices, the aliases will be propagated to all director ports to which the devices are mapped. Devices within the range that are not mapped will be skipped. If any devices in the range are mapped to a different CU image than the first device, an error will be returned. If the device range has base addresses with gaps, the aliases will also have gaps.

Mainframe ports expect devices to be mapped in groups to form CU images. The first digit in the address is the CU image number, which can range from 0 to 0xF. The remaining two digits can range from 00 to 0xFF.

To assign a PAV alias address range to a CU image, use the following form:

```
add pav alias_range nnnnn:nnnnn to mvs_ssid = nnn;
```

To assign PAV alias addresses to devices mapped to EA or EF ports, use one of the following forms, according to the version of Enginuity:

- For Enginuity version 5671, use the following form:

  ```
  add pav alias to dev SymDevName[:SymDevName], starting
  alias=cuu_address;
  ```

- For Enginuity version 5773 through 5876, use the following form:

  ```
  add pav alias to dev SymDevName[:SymDevName], alias count=nnn;
  ```

IMPORTANT

In order for these assignments to take affect, the `add pav alias_range` command must be run first, then run the `add pav alias to dev` command.

For example, to add a PAV alias range to a CU image using SSID 140, enter:

```
add pav alias_range addr 00080:0009f to mvs_ssid=140;
```

For example, to add the alias `A60` to device `01D`, which is already mapped, enter:

```
add pav alias to dev 01D starting alias=A60;
```

For example, to add total PAV aliases to device 37a, enter:

```
add pav alias to dev 37a , alias count=32;
```

Removing PAV alias addresses from mapped devices

To remove PAV alias addresses from devices mapped to EA or EF ports, use the following forms, according to the version of Enginuity:

- For Enginuity version 5671, use the following form:

```
remove pav alias from dev SymDevName[:SymDevName],
starting alias=cuu_address;
```

- For Enginuity version 5773 through 5876, use the following form:

```
remove pav alias from dev SymDevName[:SymDevName], alias count=nnn;
```

To remove a PAV alias address range from a CU image, use the following form:

```
remove pav alias_range from mvs_ssid =nnn
```

Unmapping devices

You can unmap devices from front-end director ports. This is part of the Device Reallocation feature.

To unmap devices from a director port, use the following form:

```
unmap dev SymDevName[:SymDevName] from dir
<ALL:ALL | ALL:PortNum |
DirectorNum:ALL | DirectorNum:PortNum>
[, emulation=EmulationType]
[, devmask_access=remove | retain];
```

Where:

`emulation` — Indicates the device's emulation type. This option is required when performing operations on a Celerra device, and indicates that you are aware that you are changing the Celerra environment.

`devmask_access` — This option is for Enginuity 5773 and lower. Indicates whether the device masking database should be updated. The `remove` option indicates that device masking access entries for the device should be removed from the VCMDB. The `retain` option specifies that entries remain in the database.

Note: For arrays running HYPERMAX OS 5977, the `unmap dev` command returns a feature not supported error when unmapping devices from ACLX enabled front end ports. In addition, the valid range for *PortNum* is extended to 0 to 31.

Unmapping can be from one or all ports. Since all devices with the same SSID must either be mapped or unmapped, you must provide an SSID when unmapping only some devices in a CU image.

To unmap a range of devices from EA or EF ports, use the following form:

```
unmap dev SymDevName:[SymDevName] from dir
< ALL:ALL | ALL:PortNum | DirectorNum:ALL | DirectorNum:PortNum
[new_ssid=nnn];
```

For example, to unmap five devices from a CU image so that meta CKDs can replace them, and assign the devices an SSID different from the CU image, enter:

```
unmap dev 0013B:0013F from dir all:all, new_ssid=0160;
```

Impact on I/O

The following items describe how to avoid impacting I/O.

- When unmapping, no I/O activity is allowed on any devices in the specified mapped path. Devices must be made Not Ready or Write Disabled.

  For example, to make the device Not Ready:

  ```
  symdg create -type [REGULAR | RDF1 | RDF2] DgName
  symdg -g DgName -sid SymmID add dev SymDevName
  symdg -g DgName not_ready
  ```

  If you are unmapping only one path to a multipathed device, you may prefer to write disable that path only:

  ```
  symdg -g DgName -SA 16A -p 0 write_disable
  ```

  You can make a device Not Ready without creating a device group:

  ```
  symdev -sid SymmID not_ready SymDevName
  ```

  Note: Do not use the `write_disable` argument with the `symrdf` command, as this write disables the source (R1) device(s) or the target (R2) device(s) to its/their local hosts.

- After committing a `symconfigure` mapping operation, update the device mapping information within the host system environment. Attempting host activity with a device after it has been removed or altered, but before you have updated the host's device information, can cause host errors.

  To update your hosts, run the utilities specified for your platform as described in "Introducing devices to a host". After the host environment is updated, I/O activity can resume with the array device.

Introducing devices to a host

After you reconfigure an array by moving, deleting, adding, or modifying one or more devices, you must update the host so that the host recognizes the new array configuration. For some platforms, the `symcfg scan` command is available to perform the host update. These include Sun Solaris, HP-UX, IBM AIX, Tru64/OSF1, and Windows systems.

After mapping devices to a host or changing device channel addresses, the following actions are required to introduce devices for each of the following host types.

Sun Solaris systems

To add or modify devices while online in the Solaris environment, perform the following steps:

1. Execute the following utilities:

   drvconfig or **symcfg scan**
   `disks`
   `devlinks`

2. Follow the instructions in the Solaris documentation to introduce new devices to the host environment.

HP-UX systems

To view mapping change results for HP-UX hosts, use the `ioscan` command in a statement similar to the following:

```
ioscan -fnC disk
```

You can define newly connected physical volumes to the HP-UX host system without rebooting it, using the following form:

```
insf -e
```

For more information, refer to the HP 9000 documentation.

IBM AIX systems

For AIX hosts, perform the following actions to introduce new devices:

1. From the **SMIT** menu, select **Devices › Fixed Disk › Add a Disk**.
2. Select the **EMC SYMMETRIX** definition from the disk table.
3. Select the SCSI bus on which the new disk resides.
4. Type the connection address for the new device (target, LUN).
5. Select **EXECUTE**.
6. Repeat steps 2 through 5 for each new device being added to the configuration.

HP Tru64 UNIX systems

For Tru64 UNIX hosts, perform the following actions to introduce new devices:

1. At the prompt, type:

   `scsimgr -scan_bus bus=BUSNUM`

2. Repeat for each LUN:

   a. Write a label to the device you are defining:

   `disklabel -rw rz<lun_letter><unitID> <label>`

   b. Change the ownership on the device to a particular application:

   `chown <owner>:<group> *rz<lun letter><unitID>*`

3. Follow the host documentation to introduce new devices to the host environment.

Windows systems

To introduce new or changed devices for Windows hosts, while the system remains online:

1. From the desktop, select Start, Settings, Control Panel, Add /Remove Hardware. Complete the wizard to discover and add the new devices.
2. Partition and format the new devices as described in the documentation for the specific Windows OS version.

Configuring dynamic RDF-capable devices (Enginuity 5773)

Note: Dynamic SRDF is enabled by default on arrays running Enginuity 5874 and higher. This section applies to arrays running Enginuity 5773 and lower.

Devices can be configured with `symconfigure` to be dynamic RDF-capable devices. Dynamic SRDF functionality enables you to create, delete, and swap SRDF pairs while the array is in operation. Using dynamic SRDF technology, you can establish SRDF device pairs from non-SRDF devices, and then synchronize and manage them in the same way as configured SRDF pairs.

In addition, when enabled, you can dynamically swap the SRDF device designations of a specified device group using the `symrdf` command. Source R1 device(s) become target R2 device(s) and target R2 device(s) become source R1 devices. Swaps using dynamic SRDF perform faster.

Note: Dynamic swap is not supported if Enginuity versions are mixed between your local and remote arrays.

For information on using the devices with SRDF commands, refer to the *EMC Solutions Enabler SRDF Family CLI User Guide*.

The following steps must be performed before dynamic SRDF operations can be performed:

1. The array must have the dynamic SRDF feature enabled.
2. The devices on which you want to perform dynamic SRDF operations must have the dynamic SRDF attribute set.

These actions must be performed on both the local and remote arrays.

Enabling dynamic SRDF on the device

Identify the non SRDF devices that you want available for use as dynamic SRDF devices. Create a command file that enables the dynamic SRDF operations for those devices. The following are possible attributes:

```
dyn_rdf        (Using the dyn_rdf option provides the most flexibility)
dyn_rdf1_only *
dyn_rdf2_only *
```

* These options limit a device to either an R1 or R2 device, and therefore, using them prevents the ability to perform SRDF swaps.

The following example shows such a file:

```
set dev 00020:0002A attribute=dyn_rdf;
set dev 00040:00045 attribute=dyn_rdf;
set dev 00056:0005A attribute=dyn_rdf;
```

Then, activate the option using the following command:

```
symconfigure -sid 5605 -file set_device_dyn_rdf.cmd -noprompt -v
commit
```

For example, to view the set of devices currently available for use in dynamic SRDF operations:

```
symdev list -dynamic
```

For information on using the devices with SRDF commands, refer to the *EMC Solutions Enabler SRDF Family CLI User Guide*.

Setting SRDF group attributes

RDF group attributes allow you to assign priorities to SRDF/A sessions, and to set the minimum amount of time before attempting an SRDF/A cycle switch. You can set RDF attributes for arrays running Solutions Enabler using the `symrdf` command. Refer to the *EMC Solutions Enabler SRDF Family CLI User Guide* for more details.

NOTICE

Starting with Solutions Enabler V8.0.1, the ability to set SRDF group attributes using the `symconfigure` command is no longer available. You can set RDF attributes for arrays running Solutions Enabler using the `symrdf` command.

Swapping RA groups

You can set swap devices in an RA group from target to source using the `symrdf` command. Refer to the *EMC Solutions Enabler SRDF Family CLI User Guide* for more details.

NOTICE

Starting with Solutions Enabler V8.0.1, the ability to swap SRDF group attributes using the `symconfigure swap ra group` command is no longer available. Since `symconfigure` support for performing application-specific functions requires more restrictive access and

authorization rights, generally requires more system resources (due to the session management overhead) and is more time-consuming, this functionality is no longer supported.

Converting directors (Enginuity 5876 only)

Note: This operation is not supported for arrays running HYPERMAX OS 5977 and will return the error SYMAPI_C_FEATURE_NOT_AVAILABLE.

With Enginuity 5876, you can change the emulation type of a director. This support is for FA and RA director types in a switched SRDF environment. Other director types are not supported. Use the `convert dir` option with the `symconfigure` command to convert an FA director to an RF director, and vice versa, as follows:

```
convert dir DirNum to type = <RF | FA>;
```

Note: The system checks for Access Control permission CFGSYM and for an SRDF license.

Example To convert a director type from FA to RF, create a command file containing the following command:

```
convert dir 7H to type = RF;
```

Executing the command file will convert the director type.

After the `convert director` request executes successfully, issue a `symcfg list` to verify the new director types.

FA to RF restrictions

The following restrictions apply when converting from FA to RF:

- The array must not be in SRDF/DM (data mobility) mode.
- The array must have an SRDF license to convert from FA to RF. However, you can convert from FA to RF without RDF set up on the array.
- The FA director cannot have mapped devices.
- The FA / RF conversion must be a standalone request. You cannot execute other commands (such as unmapping devices) and a director FA/RF conversion in the same request.
- Director types FA and RF are the only supported types for conversion.
- Solutions Enabler does not enforce director order (allow 7G to be RF if 7H is not).
- Conversion operations on multiple different directors will be allowed in the same `symconfigure` session, however you cannot convert the same director multiple times in the same session.
- Solutions Enabler does not support converting the director type of both directors in an SRDF pair in a single session. Separate commands must be issued for converting the directors on local and remote arrays.

- If an `abort` action is requested when the `commit` is in processing, the abort may fail if the processing has passed the point of no return.
- Conversion is not allowed when the FA director is a 16 Gb director.

RF to FA restriction

- Solutions Enabler will check in the case of RF to FA conversion that the RF director does not have any configured SRDF groups (static or dynamic).

Port to director emulation support (HYPERMAX OS 5977)

For arrays running HYPERMAX OS 5977 there is only a single emulation instance of a specific type (FA, DA, RF, EF, etc.) available per director board. If you need more connectivity, you can add additional ports to an existing emulation instance. That instance uses all cores configured to it to drive the workload across all ports assigned to it.

Each director board can contain up to 32 physical ports. These physical ports can be assigned to compatible emulation instances (and are numbered from 0 - 31) based on the rules outlined below. In addition, directors containing a Fibre Channel emulation have 32 virtual ports (numbered 32 - 63), which are reserved for use by internal guests. A capability attribute on each physical port determines the set of front-end emulations to which the port may be assigned.

You can associate (assign) unused ports to front-end emulations and disassociate (free) ports from the following emulation types:

- FA
- RF

Arrays running HYPERMAX OS 5977 do not support association of ports to EF (FICON) emulations.

Associating ports to director emulations

The `symconfigure associate port` command allows you to associate one or more physical ports to a director emulation type in the specified director board.

```
associate port <port_num>[,<port_num>. . .] to dir <dir_num>;
```

Where *dir_num* is director number of the emulation.

For example, to associate ports 2 and 4 to emulation type FA on director 7E, use the following command:

```
associate port 2,4 to dir 7E ;
```

Executing the command file assigns ports 2 and 4 to FA emulation on director 7E. After successfully assigning the ports to a given emulation on a director, issue the `symcfg list -dir ALL` command to verify that the ports are assigned to the requested director.

```
symcfg list -dir ALL -sid 064

Symmetrix ID: 000197100064 (Local)

   S Y M M E T R I X    D I R E C T O R S

    Ident  Type          Engine  Cores  Ports  Status
    -----  ------------  ------  -----  -----  ------
```

```
IM-7A  IM              1      1      0    Online
IM-8A  IM              1      1      0    Online
ED-7B  EDS             2      2      0    Online
ED-8B  EDS             2      2      0    Online
DF-7C  DISK            3      2      4    Online
DF-8C  DISK            3      2      4    Online
FA-7E  FibreChannel    4      1      2    Online
FA-7E  FibreChannel    4      1      4    Online
FA-8E  FibreChannel    4      1      3    Online
RF-7G  RDF_BI_DIR      5      1      1    Online
RF-8G  RDF_BI_DIR      5      1      1    Online
```

Rules and restrictions

- The following security privileges are required to execute this operation:
  - Required Access type: CFGSYM
  - Required Authorization Rights: Storage Admin
- Only free ports can be associated with director emulations. You can display the free ports by using the CLI command `symcfg list -sid xxx -port -free`.
- Associating ports to RE and EF emulations is not supported.
- Virtual ports can not be associated or disassociated from an emulation.
- Ports cannot be added to IM, EDS, or DA emulations. If the specified director is running one of those emulations, the operation fails with the error SYMAPI_C_DIR_IS_NOT_A_FRONT_DIR.
- If any of the specified ports is already associated with an emulation, the operation fails with the error SYMAPI_C_PORT_ALREADY_ASSOCIATED and none of the specified ports will be associated.
- If the capabilities of any of the specified ports are incompatible with the emulation running on the director, the operation fails with the error SYMAPI_C_EMULATION_PORT_MISMATCH and none of the specified ports will be associated.
- Fibre Channel ports can only be associated with FA and RF emulations.
- Associating ports to director emulation configures the ports in `Offline` state. You have to change the state of the ports to `Online` to make them usable.

Disassociating ports from director emulations

The `symconfigure disassociate port` command allows you to disassociate one or more physical ports from an emulation type in the specified director board.

```
disassociate port <port_num>[,<port_num>. . .] from dir <dir_num>;
```

Where *dir_num* is director number of the emulation.

For example, to disassociate ports 1 and 4 from emulation type FA on director 7E, use the following command:

```
disassociate port 1,4 from dir 7E ;
```

After dissociating ports from a given director emulation, issue the `symcfg list -port -free` command to list the ports currently available to be associated with a director along with their supported interface types, maximum speed, and status.

```
symcfg list -port -free -sid 064

Symmetrix ID: 000197100064

              Flags  Speed
  Slot  Port  FCIS   Gb/sec  Status
  ----  ----  -----  ------  -------

     7    18  Y...        8  Powered
     7    20  .Y..        8  Powered
     8    19  ..Y.        8  Powered
     8    21  ...Y        8  Powered

Legend:
  Flags:
    (F)C       : Y = Yes, . = No
    F(C)OE     : Y = Yes, . = No
    F(I)CON    : Y = Yes, . = No
    I(S)CSI    : Y = Yes, . = No
```

Rules and restrictions

- The following security privileges are required to execute this operation:
  - Required Access type: CFGSYM
  - Required Authorization Rights: Storage Admin
- Disassociating ports from RE and EF emulations is not supported.
- Virtual ports can not be associated or disassociated from an emulation.
- Passthru ports can not be disassociated from an emulation.
- Ports cannot be removed from IM, EDS, or DA emulations. If the specified director is running one of those emulations, the operation fails with the error SYMAPI_C_DIR_IS_NOT_A_FRONT_DIR.
- If any of the specified ports is not associated with the specified director, the operation fails with the error SYMAPI_C_EMULATION_PORT_MISMATCH and none of the specified ports will be disassociated.
- If the specified director is running an FA or FE emulation and one of the specified ports is in a port group, the operation fails with the error SYMAPI_C_DIR_HAS_PORT_IN_USE.
- If the specified director is running an SE emulation and one of the specified ports has IP interface configured on it, the operation fails with the error SYMAPI_C_DIR_HAS_PORT_IN_USE.
- If the specified director is running an RF emulation and one or more of the specified ports has RDF groups defined on it, the operation fails with the error SYMAPI_C_DIR_HAS_PORT_IN_USE.
- Ports can be disassociated from director emulations only if they are in `Offline` state.

Setting port characteristics

To set the port characteristics of a specified director, use the following form:

```
set port DirectorNum:PortNum
[FlagName=enable|disable][, ...] ]
gige primary_ip_address=IPAddress
     primary_netmask=IPAddress,
     default_gateway=IPAddress,
     isns_ip_address=IPAddress
     primary_ipv6_address=IPAddress,
     primary_ipv6_prefix=<0 -128>,
[fa_loop_id=Integer] [hostname=HostName];
```

To reconfigure an existing port by copying the port flag settings of another port, use the following form:

```
set port DirectorNum:PortNum,
copying port DirectorNum:PortNum,
[fa_loop_id=Integer] [hostname=<HostName>];
```

When copying the port flag settings from one port to another, both ports must be the same type (Gig-E to Gig-E, FA to FA, and so on). When copying Gig-E ports, IP addresses will not be copied.

Note: Before you can copy the port flags from one port to another, you must first take the port being copied offline with the `symcfg offline` command.

Where:

FlagName — A SCSI or fibre port flag. Possible values for the SCSI protocol flags are in Table 17 on page 269, and the values for the fibre protocol flags are in Table 18 on page 271.

CAUTION

Incorrectly changing the port flags can render your storage system inaccessible. Be sure of your needs before resetting these flags.

gige — Indicates that one or more network address values are going to be specified for a front-end Gig-E director. Addresses should use the Internet standard dot notation.

`primary_ip_address` — The IP address for a front-end Gig-E port.

CAUTION

When changing the primary IP address, be sure to update all connected hosts with the new address.

`primary_netmask` — The IP netmask for a front-end Gig-E port.

`default_gateway` — The gateway or router address for a front-end Gig-E port.

`isns_ip_address` — The IP address for the Internet Storage Name Service (ISNS) associated with a front-end Gig-E port.

`primary_ipv6_address` — The IPv6 address for the front-end Gig-E port.

`primary_ipv6_prefix` — The IPv6 mask prefix for a front-end Gig-E port. The value can be 0 - 128, indicating the number of initial bits in the subnet that are identical.

`fa_loop_id` — The FA director loop ID (arbitrated loop physical address). (0 - 125) (Hard Addressing must be enabled.) Not applicable for Gig-E ports.

`hostname` — The 12-character hostname.

Table 17 lists the SCSI protocol flags and their descriptions.

Table 17 SCSI protocol port flags (page 1 of 2)

SCSI protocol flags	Description
Auto_Busy	When enabled specifically for Unisys A-series platforms only, this flag enables the auto-busy mechanism so that the array returns a Busy to all Unisys host requests.
Avoid_Force_Negotiate	When enabled for Sequent V4.2.3 and lower, the array never initiates negotiations. Normal array behavior is to initiate negotiations after an offline-to-online transition. This is for hosts that do not handle negotiations.
Avoid_Reset_Broadcast	When enabled, a SCSI bus reset only occurs to the port that received the reset (not broadcast to all channels).
Command_Reordering	When enabled with Tag Command Queuing in use, the incoming SCSI commands become reordered to Simple Queuing. The default is enabled and should only be disabled upon a request from EMC.
Common_Serial_Number	This flag should be enabled for multipath configurations or hosts that need a unique serial number to determine which paths lead to the same device. **Note:** This flag is not supported on arrays running HYPERMAX OS 5977. Setting this flag will return a feature not supported error.
Cyl_Count_In_Name[a]	When this flag is enabled, the array with the specified port embeds the cylinder count into the product ID returned in the SCSI Inquiry command. Enabled for Pyramid only when it is desirable to embed the array support into the Pyramid kernel.
Disable_False_Disconnect[a]	When enabled for debugging, this flag prevents the port from performing a False Disconnect operation. (Default is disabled and currently, you cannot change this flag.)
Disable_Interleaved_Cmds[a]	When enabled (always), metavolume command interleaving is being supported. This allows multiple metamembers to operate at the same time on the same volume.
Disable_Mini_Q[a]	When enabled for debugging, this flag disables the use of the Mini Queue on the port. (Default is disabled and currently, you cannot change this flag.)
Disable_Q_Reset_on_UA	When enabled, a Unit Attention (UA) that is propagated from another director does not flush the queue for this device on this director. Used for hosts that do not expect the queue to be flushed on a 0629 sense (only on a Hard Reset).
Disable_Ultra[a]	When enabled, this flag disables Ultra SCSI on an Ultra capable SA port. (Default is disabled and currently, you cannot change this flag.)
Environ_Set	When enabled, this flag enables the environmental error reporting by the array to the host on the specific port.
HP3000_Mode[b]	When enabled for HP MPE 5.0 and Enginuity version 5062 and lower, this flag causes the array port to return a SCSI Busy state instead of a 0B44 sense code when an xx3C error occurs.
Linked_Commands[a]	When enabled, this flag enables support of SCSI linked commands. It allows a host to chain SCSI commands in a manner similar to mainframe Channel Command Words (CCWs). (Default is enabled, and currently, you cannot change this flag.)

Table 17 SCSI protocol port flags (page 2 of 2)

SCSI protocol flags	Description
Negotiate_Reset	When enabled for AS/400 hosts, this flag forces a SCSI negotiation by the array after: • A SCSI reset • An error • A bus device reset This flag is used for AS/400 systems only (default is off).
PBAY_Monitor	For the Sequent platforms only to allow emulation of the Sequent PBAY. When enabled, this flag enables low-level polling of the SCSI bus in order to intercept the nonstandard SCSI operations required for a Sequent PBAY disk subsystem. Must be used for the Sequent cluster operation for the Symmetry system for Sequent V4.2.x operating systems only. Must not be used on versions higher than V4.2.x or for any NUMA-Q systems and also not used for Fibre Channel.
SCSI_3	When enabled, the Inquiry data is altered when returned by any device on the port to report that the array supports SCSI 3 protocol. When this flag is disabled, the SCSI 2 protocol is supported.
SCSI_Support	When enabled, this flag provides a stricter compliance with SCSI standards for managing device identifiers, multi-port targets, unit attention reports, and the absence of a device at LUN 0.
Sequent[b]	When enabled, the array on the specified port does not force wide or synchronous negotiations, and sets the task timeout window to be 15 seconds before aborting a process. In addition, a busy status is returned instead of an 0B44H when aborting a command on a timeout.
Server_On_AS400	When enabled for AS/400 platforms, this flag indicates that the port is to behave as a server, returning server inquiry data, rather than AS/400 data. **Note:** This flag is not supported on arrays running HYPERMAX OS 5977. Setting this flag will return a feature not supported error.
Set_Qerr	This flag should be enabled for SGI platforms only to flush the queue on a contingent allegiance condition (CAC). Must be used for V5.3 and V6.2 SGI operating systems and cluster environments. Not used on versions higher than V6.2.
Siemens[b]	For Siemens R-Series platforms only. When this flag is enabled for Siemens, the array returns in the sense data error 0B48, instead of 0B44 for normal behavior.
Soft_Reset	When enabled for a Bull/GCOS-7 host, the array port supports the SCSI Soft Reset option.
SPC2_Protocol_Version	This flag should be enabled (default) in a Windows 2003 environment running Microsoft HCT test version 12.1. When setting this flag, the port must be offline. **Note:** Reboot the host after setting this flag.
Sunapee[b]	When enabled for Sun PDB clusters, this flag enables the Sunapee option on the port.
Sync_Transfer[a]	When enabled, this flag enables SCSI synchronous negotiations. (Default is enabled, and currently, you cannot change this flag.)
Tagged_Commands[a]	When enabled, this flag enables support for tagged commands. (Default is enabled, and currently, you cannot change this flag.)
Wide_Transfer[a]	When enabled, this flag enables SCSI Wide operation. (Default is enabled, and currently, you cannot change this flag.)

a. Not available for host-based configuration changes.

b. Obsolete in Enginuity 5874 and higher.

Table 18 lists the Fibre protocol flags and their descriptions.

Table 18 Fibre protocol port flags

Fibre protocol flags	Description
ACLX	When enabled, allows storage provisioning using Auto-provisioning Groups. This flag is applicable for Enginuity 5874 and higher.
AS400	This flag should be enabled for any AS400 (iSeries) hosts connecting to the port. **Note:** This flag is not supported on arrays running HYPERMAX OS 5977. Setting this flag will return a feature not supported error.
Auto_Negotiate	When enabled, allows two fibre ports to *handshake* and settle on an optimal speed for data transfer.
Init_Point_to_Point[a]	When enabled, specifies a point-to-point (direct or switched) topology in the initialization sequence. When disabled (default), it is initialized as an arbitrated loop. **Note:** This flag is not supported on arrays running HYPERMAX OS 5977. Setting this flag will return a feature not supported error.
Non_Participating[a]	When enabled along with the Hard_Addressing flag, the Fibre Channel director only uses hard-assigned addressing when it initializes on the loop. Otherwise, soft-assigned addressing is used during loop initialization (the default).
OpenVMS[b]	Enabled for an OpenVMS fibre connection.
Unique_WWN[a]	When enabled (default) for all environment configuration changes and new environments to ensure unique World Wide Names (WWN) within the fibre environment (uses array serial numbers and port numbers). When disabled, you don't have to change WWNs.
VCM_State[c]	Enabled for device masking or the Volume Logix software, which provides volume configuration management controls to handle access to array devices. (Disabled is the default.) This flag is only applicable for Enginuity 5773 and earlier.
Volume_Set_Addressing[b]	When enabled along with the Disk_Array flag for HP-UX hosts, the volume set addressing mode is selected. VSA mode allows octal addressing.

a. Not available for Gig-E ports.

b. A block is added to prevent OpenVMS and Volume_Set_Addressing Fibre protocol port flags from being set at the same time on any given port, as setting these two flags together may result in data loss. This block will be effective for all Enginuity levels supported.

c. Obsolete in Enginuity 5874 and higher.

For example, to turn on the write protect access logix (ACLX) for director `7E`, port `0`, enter:

```
set port 7e:0 ACLX=enable;
```

When setting port attributes, it is recommended that you temporarily suspend I/O activity to the effected ports during this operation.

Setting a port to show ACLX device

For arrays running HYPERMAX OS 5977 or higher, devices can no longer be mapped to ACLX enabled ports. To make the ACLX device visible or to remove visibility from the ACLX device, the `Show_ACLX_Device` attribute can be set on front end ports to ENABLE or DISABLE. By default all ACLX enabled ports will have the `Show_ACLX_Device` attribute disabled.

For example, to make the ACLX device visible:

```
symconfigure -sid 230 commit -cmd "set port 5E:0 Show_ACLX_device=ENABLE;"

Execute a symconfigure operation for symmetrix '000197100230' (y/[n]) ? y

A Configuration Change operation is in progress. Please wait...

    Establishing a configuration change session...............Established.
    Processing symmetrix 000197100230
    {
          set port 5E:0 Show_ACLX_device=ENABLE;
    }

    Performing Access checks..................................Allowed.
     . . .
    Terminating the configuration change session..............Done.

The configuration change session has successfully completed.
```

Reporting port flag details

To report the new `SHOW_ACLX_DEVICE` port attribute configured on an array port, the list -fa -v or the `list -fa -detail` commands are used. The following is an example output reporting the port detail for an ACLX device:

```
symcfg -sid 064  list -port -fa 7e -detail

Symmetrix ID: 000197100064

       S Y M M E T R I X    D I R E C T O R    P O R T S

                                                Flags  Speed
    Ident  Port  WWN               Type         ASVPG  Gb/sec  Status
    -----  ----  ----------------  ------------ -----  ------  -------

    FA-7E     0  50000972C011C918  FibreChannel ...X.       8  Online
    FA-7E     1  50000972C011C919  FibreChannel XX...       8  Online

Legend:
  Flags:
      (A)CLX Enabled              : X = True, . = False
      (S)how ACLX device Enabled  : X = True, . = False, - = N/A
      (V)olume Set Addressing     : X = True, . = False
      (P)oint to Point            : X = True, . = False
     VNX (G)ateway Direct Attach : X = True, . = False
```

The "`symdev list-FA ALL -v`", "`sympd list -SA ALL -v`", "`symdev show`", and "`sympd show`" commands display the LUN information of an ACLX device on all ACLX enabled ports where the `Show_ACLX_Device` port attribute is Enabled.

Note: For arrays running Enginuity 5876 or lower, the Show_ACLX_Device port attribute reports as N/A.

Enable SYMAPI environment option

Environment option SYMAPI_CTRL_OF_NONVISIBLE_DEVS in the `options` file must be enabled (or not present in the options file) if there is no device from the local host mapped to this port.

CHAPTER 8
Configuring Virtual Provisioning

This chapter describes Virtual Provisioning™ concepts and how to perform operations using the SYMCLI configuration change component.

- Overview 276
- Creating DATA devices (Enginuity 5773 through 5876) 278
- Creating thin devices (Enginuity 5773 through 5876) 279
- Creating thin pools (Enginuity 5773 through 5876) 281
- Performing actions on thin devices (Enginuity 5773 through 5876) 284
- Stopping background operations on thin devices 291
- Draining and rebalancing a thin pool (Enginuity 5874 through 5876) 292
- Virtual Pool (VP) compression (Enginuity 5876 only) 296
- Monitoring thin pools 298
- Viewing thin pools 299

Overview

Virtual Provisioning allows for storage to be allocated and accessed on demand from a pool of storage servicing one or many applications. This chapter details the Virtual Provisioning operations that are available using various CLI commands.

Refer to *EMC VMAX Family Product Guide* for more information on Virtual Provisioning.

Overview for arrays running HYPERMAX OS 5977

VMAX3 arrays running HYPERMAX OS 5977 are pre-configured at the factory with virtually provisioned devices. Virtual Provisioning helps reduce cost, improve capacity utilization, and simplify storage management. Users can present a large amount of capacity to a host and then consume space only as needed from a shared pool. Virtual Provisioning ensures that thin pools can expand in small increments while protecting performance, as well as non-disruptive shrinking of thin pools to help reuse space and improve capacity utilization.

Refer to *EMC VMAX Family with HYPERMAX OS Product Guide* for more information on Virtual Provisioning for arrays running HYPERMAX OS 5977.

With VMAX3 arrays, all devices are under FAST control and FAST handles all device binding and allocation. Therefore, user controlled actions on thin pools (such as bind, unbind, rebind) are not supported on these arrays.

You can perform the following Virtual Provisiong operations on thin devices using the `symdev`, `symsg` (storage groups), `symcg` (composite groups), and `symdg` (device groups) commands:

- Allocating space in thin devices, refer to "Allocating and reclaiming space on thin devices" on page 288
- Reclaiming space on thin devices, refer to "Reclaiming allocations" on page 289
- Freeing device allocations, refer to "Freeing all allocations or written tracks (HYPERMAX OS 5977 only)" on page 291
- Renaming a thin pool, refer to "Renaming thin pools" on page 283
- Monitoring a thin pool, refer to "Monitoring thin pools" on page 298
- Viewing a thin pool, refer to "Viewing thin pools" on page 299

For arrays running HYPERMAX OS 5977, the following commands are blocked for bind, unbind, and rebind operations:

- symdev
- symdg
- symsg
- symcg

Restrictions

The following operations on DATA devices and thin pools are *not* supported on arrays running HYPERMAX OS 5977:

- Creation or deletion of DATA devices
- Creation or deletion of thin pools

- Management operations on existing DATA devices including:
  - Adding or removing DATA devices from thin pools
  - Enabling or disabling DATA devices
  - Activating or deactivating DATA devices
  - Starting or stopping the draining of DATA devices
  - Placing device reservations on DATA devices and adding DATA devices to existing device reservations
- Management operations on thin pools including:
  - Setting pool attributes
  - Starting and stopping pool rebalancing
  - Enabling or disabling VP compression on a pool

Overview for arrays running Enginuity 5773 through 5876

For arrays running Enginuity 5773 through 5876 the Virtual Provisioning feature involves managing these components: *thin devices* (TDEV), *DATA devices* (DATADEV), and *thin pools*. Thin devices can be created with an inflated capacity, because the actual storage space for the data written to the thin devices is on the DATA devices. In this way, when additional storage is needed, more DATA devices can be created in the thin pool. DATA devices are not visible to the host.

Refer to *EMC VMAX Family Product Guide* for more information on Virtual Provisioning for arrays running Enginuity 5773 through 5876

IMPORTANT

Not all Virtual Provisioning operations are supported on earlier Enginuity versions (such as 5773). Refer to "Supported configuration operations" on page 209 for supported operations for Enginuity 5773 through 5876.

You can create thin devices, DATA devices, and thin pools using the `symconfigure` command.

The following operations manage thin device binding and unbinding, and are performed using the `symdev`, `symsg` (storage groups), `symcg` (composite groups), and `symdg` (device groups) commands:

- bind, unbind, rebind thin devices to thin pools, refer to "Performing actions on thin devices (Enginuity 5773 through 5876)" on page 284
- start allocate, stop allocate, refer to "Allocating and reclaiming space on thin devices" on page 288
- free, stop free, refer to "Freeing written tracks (Enginuity 5874 through 5876)" on page 290
- reclaim, stop reclaim, refer to "Reclaiming allocations" on page 289

Note: The `symconfigure` command is no longer used to perform bind, unbind, rebind, allocate, reclaim, and free operations on thin devices and thin pools.

Solutions Enabler supports Virtual Provisioning with Open Replicator, SRDF/Synchronous, SRDF/Asynchronous, SRDF Data Mobility (SRDF/DM), TimeFinder/Mirror, TimeFinder/Clone, and TimeFinder/Snap.

The data of a thin device can be spread across multiple pools. The `symcfg` command displays all of the pools in which a thin device has allocated tracks.

Enginuity 5876 supports cascaded storage groups. Any of the virtual provisioning operations performed on a parent storage group are also performed on all devices contained by all child storage groups of a specified parent storage group.

Creating DATA devices (Enginuity 5773 through 5876)

DATA devices are similar to SAVE devices, in that they are not visible to the host and must be contained in a pool before they can be used. Thin pools can only contain devices of the same emulation and protection type, however, the DATA devices can be different sizes.

Use the following `symconfigure` command to create DATA devices:

```
create dev count=n,
  size = n [MB | GB | CYL],
  emulation=EmulationType,
  config=DeviceConfig
  [, attribute=<ckd_meta | savedev | datadev>
  [in pool=PoolName]
  [member_state=<ENABLE | DISABLE> ]
```

Where:

`create dev count` — Specifies the number of DATA devices you are creating.

`size` — Indicates the size of the DATA device. If MB or GB is not specified, the size defaults to cylinders.

`emulation` — Specifies FBA or CKD 3390 devices.

`config` — Specifies the device configuration. The following configuration types are supported for DATA devices: 2-Way-Mir, RAID5 (3+1 and 7+1), and RAID6 (6+2 and 14+2).

`attribute` — Specifies the device type datadev.

`in pool` — Specifies the name of the thin pool.

`member_state` — Indicates the state of the DATA device; enable or disable. If a device is disabled, it cannot receive writes.

DATA device configuration rules

The following configuration rules apply to DATA devices:

- A disk group must be specified when creating a DATA device if the `create dev` command is also adding the device(s) to a pool.
- The only supported emulation types are FBA and CKD 3390.
- There is no default thin pool.
- A DATA device cannot be replicated (no BCV, RDF, and so on).

- A DATA device cannot be a metadevice.
- DATA devices are not visible to the host; they cannot be mapped or masked.
- RAID 10 protection for CKD 3390 DATA devices is not supported.

Creating thin devices (Enginuity 5773 through 5876)

Thin devices are devices that may or may not have storage allocated to them when they are created. To a host operating system, they look like regular devices with their configured capacity. The host treats them as regular devices and writes and reads from these devices like regular devices.

Use the `symconfigure` command to create thin devices. Allocating persistent tracks during the creation of a thin device (binding a thin device to a pool) and during the allocation of a thin device is supported. In addition, reclaiming allocated persistent tracks is also supported.

For thin metadevices, refer to "Configuring thin metadevices (Enginuity 5773 through 5876)" on page 235.

Use the following syntax to create thin devices:

```
create dev count=n,
  size = n [MB | GB | CYL],
  emulation=EmulationType,
  config=DeviceConfig
  [, remote_config=DeviceConfig,
  [, binding to pool=PoolName
  [preallocate size = ALL]
  [allocate_type=persistent]]
  [remote_pool=PoolName]];
```

Where:

`create dev count` — Specifies the number of thin devices you are creating.

`size= n [MB |GB |CYL]` — Specifies the size of each thin device. If MB or GB is not specified, the size defaults to cylinders.

`emulation` — Indicates FBA, CKD 3390, or AS/400 D910_099 devices.

Note: Thin CKD devices require Enginuity V5876 or higher.

`config` — Specifies TDEV (local thin device), RDF1+TDEV (local SRDF thin device), RDF2+TDEV (remote SRDF thin device), or BCV+TDEV.

`remote_config` — Specifies RDF2. The `remote_config` option is only used when creating both local and remote SRDF thin devices.

`binding to pool` — Specifies the name of the local thin pool.

`preallocate size` — The only preallocate size is ALL, which is the entire device. The device size must be less than or equal to the available space in the pool. This option is only used when a newly created device is also being bound to a pool.

`allocate_type` — With this optional parameter, the allocations are unaffected by any reclaim operations, as well as clone, snap, or SRDF copy operations. If it is not supplied, `symconfigure` will preallocate non persistent tracks which can be reclaimed without any additional reclaim flags.

Note: To remove the persistent allocation, unbind the thin device or execute a `symconfigure start reclaim on dev` command with a persistent type qualifier.

`remote_pool` — Specifies the name of the remote thin pool.

Example for creating thin devices

To create ten 2 GB thin devices, bind them to thin pool `tp_pool`, preallocate 1 GB to each thin device, and mark the allocations as persistent, enter:

```
symconfigure commit -sid 12345 -file add_new_tdevs.cmd
```

Where `add_new_tdevs.cmd` contains:

```
create dev count=10, size=2GB
  emulation=FBA, config=TDEV,
  binding to pool tp_pool, preallocate size=ALL
  allocate_type=persistent;
```

The same metadevice commands (`form meta`, `dissolve meta`, `convert meta`) can be used to configure a thin metadevice. Table 19 provides the supported thin metadevice features.

Table 19 Supported thin metadevice features

Metadevice feature	Thin device support
Form a thin meta	• Only thin devices can be used to create a thin metadevice. Mixing of thin and non thin metadevices to create a metadevice is not allowed. • Only unbound thin devices can be formed into a metadevice. • For a striped metadevice, all members should have the same device size.
Dissolve a thin meta	Only unbound thin metadevices can be dissolved.
Expand a thin meta	A bound striped or concatenated metadevice can be expanded; however, the new meta member must be unbound. As part of the expansion process, the new metamember will be bound to the same thin pool. The `protect_data` option is not supported for concatenated metadevices.
Convert a thin meta	Concatenated thin metadevices cannot be converted to striped metadevices. See Table 16 on page 238 for supported thin device conversion types.

Thin device configuration rules

The following rules apply to thin devices:

- The supported emulation types are FBA, CKD 3390, and AS/400 D910_099.
- Thin devices can only be bound to one pool.

- The thin pool to which the thin devices are bound must have free tracks.
- A thin device inherits its protection type from the pool to which it is bound. Whatever configuration type the DATA devices have, the thin devices have.
- Many thin devices can use the same pool.
- PowerPath® consistency groups are not supported for thin devices.
- The only supported SRDF configuration is when both the R1 and the R2 are thin devices.
- For CKD thin devices (VP CKD), the devices must be fully allocated (`preallocate size=ALL`) with the persistent option specified (`allocate_type = persistent`).

Thin devices can be converted to other thin device configurations. Table 16 on page 238 lists the valid conversions.

Creating thin pools (Enginuity 5773 through 5876)

Thin pools can be created at the same time as creating the DATA devices. The same naming conventions that apply to SAVE pools and RDF/DSE pools apply. However, there is no default thin pool. Use the `symconfigure` command to create thin pools.

To create a thin pool, use the following syntax:

```
create pool PoolName, type = thin
[, max_subs_percent= <n>]
[, rebalance_variance = n],
[, max_dev_per_rebalance_scan = n]
[, pool_resv_cap = n]
[, vp_compression = <ENABLE>];
```

Where:

`type` — Specifies the type of device pool. Snap pools are for TimeFinder/Snap operations, SRDF/A DSE pools are for SRDF/A operations, thin pools are for Virtual Provisioning operations with thin devices.

`PoolName` — Specifies the name of the device pool. It can be from 1 to 12 alphanumeric characters long and include hyphens (-), and underscore (_) characters.

`max_subs_percent` — Specifies the maximum total capacity that can be bound to the pool. This can have a value between 0 and 65534. A value of 100 indicates that the total capacity of the thin devices bound to a pool cannot exceed 100 percent of the actual capacity of the pool. A value of 200 indicates that the total capacity of the thin devices bound to a pool cannot exceed twice the amount of the actual capacity of the pool.

A value of 1 will allow only a maximum thin capacity that totals 1/100th the actual physical capacity of the pool. The largest number 65534 indicates that the total capacity of the thin devices bound to a pool can be 65534 percent of the total physical capacity of the pool.

`rebalance_variance` — Specifies the value (in %) for the pool Rebalance Variance. It can be set in the range of 1 to 50. The default value is 1.

`max_dev_per_rebalance_scan` — Specifies the maximum limit (in %) for the pool rebalance scan device range. It can be set in the range of 2 to 1024. Its default value is 256.

`pool_resv_cap` — (FAST use only) The pool reserved capacity (PRC) is a percentage of the capacity of the virtual pool that will be reserved for non-FAST activities. If the free space in this pool (as a percentage of pool-enabled capacity) falls below the PRC, the FAST controller does not move any more chunks into the pool. Valid PRC values range from 1 to 80 percent.

Note: PRC values set at the pool-level override any globally set PRC values. For setting a global PRC value for all thin pools, refer to "Setting FAST control parameters" on page 189.

`vp_compression` — This option enables compression on the thin pool. This is only a capability setting. Data compresses or uncompresses after a command at the device or group level. Virtual Provisioning compression is explained in "Setting pool attributes (Enginuity 5773 through 5876)" on page 283.

Note: When creating thin pools it is recommend that you set up thin pool monitoring with the event daemon (`storevntd`) to allow for monitoring of pool out-of-space conditions that could result in application outages.

The maximum number of device pools per array is 512.

Thin pool restrictions and rules

The following restrictions apply to thin pools:

- The last DATA device cannot be removed from a pool if the pool is enabled for compression.
- DATA device pools must have the same disk type (EFD, FC, SATA).
- DATA devices created from external provisioning can only be added to pools of devices created from external provisioning.
- A thin pool can only contain DATA devices with one emulation type. The pool itself is not defined to be a pool for an emulation type. Instead, the first DATA device added to the thin pool defines its emulation type.
- All DATA devices in a thin pool must have the same protection. For example, if a thin pool has the first DATA device defined as a 2-Way-Mir, all subsequent added DATA devices must be 2-Way-Mir devices.
- A DATA device of a specific protection and emulation can be put only into an empty thin pool, or into a pool that *only* contains DATA devices that have the same protection and emulation type.
- A DATA device can only be removed from a thin pool if it is disabled and has no used tracks on it.
- DATA devices cannot be disabled and moved from a thin pool in the same session.

- If the total pool capacity needed to store existing thin device data will be insufficient as a result of a `disable dev SymDevStart:SymDevEnd` command, none of the devices in the requested range will be disabled.

Renaming thin pools

To rename a thin pool, use the following syntax:

```
rename pool PoolName to NewPoolName
```

Setting pool attributes (Enginuity 5773 through 5876)

You can configure the following attributes for a thin pool using the `symconfigure` command:

- **Maximum subscription percent** — This percent is the maximum limit for the thin pool. This ratio is the total host-perceived capacity of thin devices, divided by physical thin pool capacity.
- **Pool rebalancing attributes** — Pool Rebalancing is an operation in which uneven usage of data devices on a pool is evened out. With Enginuity version 5875, the `symconfigure` CLI will allow following rebalancing attributes to be set/reset on any thin pool:
  - **Rebalancing variance** — The rebalancing variance is the target device utilization variance for the rebalancing algorithm. The rebalancing algorithm attempts to level distribution of data in a pool so that the percentage utilization of any device in the pool is within the target variance of the percentage utilization of any other device in the pool.
  - **Maximum rebalance scan device range** — The maximum rebalance scan device range is the maximum number of devices in a pool on which the rebalancing algorithm will concurrently operate.
- **Pool reserved capacity** — The pool reserved capacity (PRC) is a percentage of the capacity of the virtual pool that will be reserved for non-FAST activities. If the free space in the pool (as a percentage of pool-enabled capacity) falls below the PRC, the FAST controller does not move any more chunks into the pool.
- **Compression** — Compression can be enabled when the thin pool is created, or as an attribute on an existing thin pool. When a thin pool is enabled for compression, the data in the pool can be compressed and uncompressed. "Virtual Pool (VP) compression (Enginuity 5876 only)" on page 296 provides more details.

  Note: VP compression is only supported on arrays running Enginuity 5976. VP compression of CKD devices is not supported on VMAX 10K arrays.

Note: Rebalancing and compression can only be performed on thin pools and is not applicable for Snap or RDFA/DSE pools.

When creating a thin pool, the pool attributes can be set using the following `symconfigure` command syntax:

```
create pool PoolName,
    type = <snap | rdfa_dse | thin>
    [, max_subs_percent= <n>]
```

```
[, rebalance_variance = <n>]
[, max_dev_per_rebalance_scan = <n>]
[, pool_resv_cap = n]
[, vp_compression = <ENABLE>];
```

Pool attributes can be set after the thin pool is created using the following syntax:

```
set pool PoolName,
    type = <thin>,
     <[max_subs_percent= <n | NONE>]
      [, rebalance_variance = <n>]
      [, max_dev_per_rebalance_scan = <n>]>
      [, pool_resv_cap = n | NONE]
      [, vp_compression = <ENABLE | DISABLE>];
```

The following example shows the command and output that displays when enabling compression on a thin pool:

```
symconfigure -sid 397 -nop -v -cmd "set pool poolTest, type = thin, vp_compression
   = ENABLE;" commit

A Configuration Change operation is in progress. Please wait...

    Establishing a configuration change session...............Established.
    {
      set pool poolTest type=thin, vp_compression=ENABLE;
    }

    Performing Access checks..................................Allowed.
    Checking Device Reservations..............................Allowed.
    Committing configuration changes..........................Reordering.
    Setting pool attributes ..................................Done.
    Committing configuration changes..........................Committed.
    Terminating the configuration change session..............Done.
The configuration change session has successfully completed.
```

Performing actions on thin devices (Enginuity 5773 through 5876)

Refer to Table 7, "Configuration change operations," on page 209 for a list of allowable actions on thin devices for specific Enginuity versions.

The `symdev`, `symdg` (device groups), `symsg` (storage groups), and `symcg` (composite groups) commands support the following actions when issued against thin devices: binding, unbinding, and rebinding, allocating, reclaiming, and freeing. The operations can be directed at a single device or a list of devices.

Note: For Enginuity versions 5876 or lower, free and reclaim actions are blocked for any device which is the source or target of a Clone, Emulation, or Snap session.

Use the following command syntax to perform control actions on thin devices using `symdev, symdg, symsg, or symcg command`:

```
symdev -sid SymmID [-devs <<SymDevStart>:<SymDevEnd> | <SymDevName>
               [,<<SymDevStart>:<SymDevEnd> | <SymDevName>>...]>]

          bind -pool <PoolName>

          rebind -pool <PoolName>

          unbind -pool <Poolname>
```

```
         allocate [-persistent | -stop]

         free | -stop]

         reclaim [-persistent | -stop]

symdg -dg DgName [-noprompt] [-i interval][-c count]
              [-bcv [-hop2] | -rbcv | -brbcv | -rrbcv |
              -vdev [-hop2] | -rvdev | -tgt [-hop2] |
             -rtgt] [-rp] [-star] [-celerra]

         bind -pool <PoolName>

         rebind -pool <PoolName>

         unbind -pool <Poolname>

         allocate [-persistent | -stop]

         free | -stop]

         reclaim [-persistent | -stop]

symsg -sg <SgName> -sid <SymmID> [-i <Interval>]
          [-c <Count>] [-v] [-noprompt] [-rp] [-star]
          [-celerra] [-skip -lockid <lockNum>]

         bind -pool <PoolName>

         rebind -pool <PoolName>

         unbind -pool <Poolname>

         allocate [-persistent | -stop]

         free | -stop]

         reclaim [-persistent | -stop]
symcg –cg <CgName> [-i <Interval>] [-c <Count>] [-v]
          [-noprompt] [-force]
          [-bcv [-hop2] | -rbcv | -brbcv | -rrbcv |
           -vdev [-hop2] | -rvdev | -tgt [-hop2] | -rtgt ]
          [-rp] [-star] [-sid <SymmID> |
          -rdfg <<SymmID>:<<GrpNum>[,<GrpNum>,...]|all>[,...] |
            name:<RdfGroupName>[,<RdfGroupName>]>] [-celerra]

          bind -pool <PoolName>

          rebind -pool <PoolName>

          unbind -pool <Poolname>

          allocate [-persistent | -stop]

          free [-all | -stop]

          reclaim [-persistent | -stop]
```

A file can be used to specify the list of devices to be acted upon. The file must contain one SymDevName per line. When using a file the syntax is `-file <FileName>`.

Binding thin devices

A `bind` action associates a thin device to a thin pool.

The following command binds thin devices in device group `MyDg` to the thin pool `Mythinpool`.

```
symdg -sid 234 -nop bind -dg MyDg -pool Mythinpool
```

The `symsg` and `symcg` commands are the same as `symdg` command using -sg *`SgName`* and the -cg *`CgName`*, respectively.

The following command binds thin devices 1078 - 1081 to the thin pool `Mythinpool`.

```
symdev -sid 234-nop bind -devs 1078:1081 -pool Mythinpool
```

Note: AS/400_D910_099 devices can only be bound to FBA thin pools that do not contain externally provisioned DATA devices.

Unbinding thin devices

An `unbind` action disassociates a thin device from a thin pool.

For example, the following commands unbinds thin devices in device group `MyDg` from the thin pool `Mythinpool`:

```
symdg -sid 234 -nop unbind -dg MyDg -pool Mythinpool
```

The `symsg` and `symcg` commands are the same as `symdg` command using -sg *`SgName`* and the -cg *`CgName`*, respectively.

The following command unbinds thin devices 1078 - 1081 from the thin pool `Mythinpool`:

```
symdev -sid 234-nop unbind -devs 1078:1081 -pool Mythinpool
```

Restrictions for unbinding thin devices

The following are restrcitions for unbinding thin devices:

- Device cannot be mapped to a front-end port or in the Ready state
- Device cannot be masked
- Device cannot have active snap sessions
- Device cannot be held
- Device cannot be a source or target of a clone (src or tgt) session
- Device cannot be a metamember
- Device cannot be a part of an enabled SRDF CG group
- Device cannot be an SRDF device
- Device cannot be under FAST control

Rebinding thin devices

A `rebind` action allows you to rebind a thin device to a new pool without moving its data or losing any data. Rebind simply changes the thin device's current binding to a new pool. Rebind will not move any existing data to the new pool, but all new allocations to the thin device will go to the new pool after the rebind action completes.

For example, the following commands rebinds thin devices in device group `MyDg` to the new thin pool `Mythinpool2`.

```
symdg -sid 234 -nop rebind -dg MyDg -pool Mythinpool2
```

The `symsg` and `symcg` commands are the same as `symdg` command using -sg *`SgName`* and the -cg *`CgName,`* respectively.

The following command binds thin devices 1078 - 1081 to the new thin pool `Mythinpool2`.

```
symdev -sid 234-nop rebind -devs 1078:1081 -pool Mythinpool2
```

Restrictions for rebinding thin devices

The following are restrictions for rebinding thin devices:

- A thin device must be in the Bound state before issuing a rebind.
- A rebind operation is not allowed for devices that are the target of a VP Snap session.
- The new binding has to comply with the over subscription ratio of the new pool. The entire size of the device being rebound will be considered when calculating the over subscription.
- If devices in a range, device group, or storage group are bound to different pools, then all the devices will be rebound to the specified pool. No warning/errors will be issued.
- If a thin device is part of a storage group that is under FAST management, the thin device can only be bound to a pool in a tier that is part of the FAST policy associated with the storage group. Therefore, the device can only be rebound to a pool that is within the policy.
- If all the devices that are being rebound are already bound to the destination pool, an error returns. If some devices get bound to a pool different than what they are currently bound to, the CLI operation will return a success status.

For example, use the `symdev show` command to check that the pool listed is the pool to which the device is bound, as follows:

```
symdev show -sid 351 1078

    Device Physical Name     : Not Visible

    Device Symmetrix Name    : 01078
    Device Serial ID         : N/A
    Symmetrix ID             : 000194900351

    Number of RAID Groups    : 0

    Attached BCV Device      : N/A

    Attached VDEV TGT Device : N/A

    Vendor ID                : EMC
    Product ID               : SYMMETRIX
    Product Revision         : 5876
```

```
Device WWN               : 60000970000194900351533031303738
Device Emulation Type    : FBA
Device Defined Label Type: N/A
Device Defined Label     : N/A
Device Sub System Id     : 0x0011
Cache Partition Name     : DEFAULT_PARTITION
Bound Pool Name          : pool_thin_1

Device Block Size        : 512

    . . .
```

Note: Bound Pool Name will be reported as N/A for an encapsulated device.

Allocating and reclaiming space on thin devices

The `allocate` action allocates storage for specified devices. The `-persistent` option is used to specify that allocated storage cannot to be reclaimed or freed. To reclaim allocated storage use the optional `- persistent` option with the `reclaim` action.

If any of the tracks specified for persistent allocation are already allocated, the already allocated tracks will be marked as persistent. The allocation is performed on track groups of 12 tracks (for HYPERMAX OS 5977 track group size is 1 track). Therefore if any tracks in the track group are persistent, the whole track group becomes persistent, even if some part of the track group was allocated by a command that did not request a persistent allocation.

Note: Allocating and reclaiming space on devices is supported for HYPERMAX OS 5977 and all supported Enginuity versions. The `free` action is supported on Enginuity 5874 through 5876, and the `free -all` action is only supported on HYPERMAX OS 5977.

For example, the following command starts allocation on the thin devices in the device group `myDg`:

```
symdg -sid 234 -nop -v allocate -dg myDg -persistent
```

The `symsg` and `symcg` commands are the same as `symdg` command using -sg *`SgName`* and the -cg *`CgName`*, respectively.

The following command starts allocation on thin device 1078 - 1081:

```
symdev -sid 234 -nop -v allocate -devs 1078:1081 -persistent
```

Note: Actions to allocate, free, or reclaim thin devices are performed as background operations.

Restrictions

The following restrictions apply to allocating action:

- If any of the devices in the device group or storage group are in a state other than allocating or bound, the command will fail.
- If a device group is specified, the action of the command is limited to the standard devices in the device group only. Space reclamation

- Unwritten storage can be recovered with either a `free` or a `reclaim` action. Allocations that were written as zeros can only be recovered by using the `reclaim` action. Written tracks are freed with the `free` action. Additionally, storage that was allocated with the persistent attribute can only be recovered by using the `-persistent` option with the `reclaim` action.

Reclaiming allocations

Reclaiming persistent allocations frees up tracks that are unwritten or zero-based, even if they are marked as persistent. This is the only way that persistent tracks can be freed using this command.

The command returns a success status if there are no allocations on the specified thin device.

If `-persistent` is not specified, this command frees up both unwritten tracks and those that are completely written with zeros. It will not free up tracks that are marked persistent.

For example, the following command starts reclaiming space on thin devices in the device group `myDg`:

```
symdg -sid 234 -nop -v reclaim -dg myDg -persistent
```

The `symsg` and `symcg` commands are the same as `symdg` command using -sg *`SgName`* and the -cg *`CgName,`* respectively.

The following command starts reclaiming space on thin device 1078 - 1081:

```
symdev -sid 234 -nop -v reclaim -devs 1078:1081 -persistent
```

Restrictions for using reclaim

- If a device group is specified, the action of the command is limited to the standard devices in the device group only.
- The reclaim operation is ignored for any non-thin devices in a range, device group, or storage group.
- If the devices in a device group or storage group are bound to different pools, the reclaim operation is allowed on the device group or storage group.
- Space reclamation cannot be performed while a DATA device in the pool is draining.

For Enginuity versions 5876 or lower, the reclaim action is blocked for:

- Any device which is the source or target of a TimeFinder/Clone or Snap session when the state is not copied or restored.
- Any device which is the source or target of a Emulation session when the state is not split. The reclaim action is allowed if the device state is split, but it cannot be a background split. The flags in the `symmir list` command report if source or target device is in a background split, as shown in the following example:

```
symmir list -sid 3264
Symmetrix ID: 000000003264

Standard Device            BCV Device            State
-------------------- ----------------------- --------------
    Invalid    Invalid
Sym Tracks Sym Tracks    GBE                   STD <=> BCV
-------------------- ----------------------- --------------
002B  0    0E0B     0     ...                  Synchronized
```

```
002E  0    0E00     0    ..X                    Synchronized
002E  0    0E0A     0    ...                    Synchronized
0032  0    0E0F     0    ...                    Split
00FF  0    00FD     0    ...                    Split
0DF5  0    0DA5     0    ..X                    Synchronized
0DF5  0    0DA4     0    ..X                    Synchronized
0F70  0    001B   3592    X..                   SyncInProg
0F71  0    001C   4496    X..                   SyncInProg
0F93  0    0DF9     0    ..X                    Split
1015  0    1069     0    X..                    Synchronized

Total
      -------- --------
Tracks   0       8088
MB(s)    0.0      505.5

Legend:

(G): X = The paired BCV device is associated with a group.
. = The paired BCV device is not associated with a group.
(B): X = The paired BCV device is splitting in the background.
. = The paired BCV device is not splitting in the background.
(E): X = The paired BCV device is emulation mode.
. = The paired BCV device is not emulation mode.
```

Freeing written tracks (Enginuity 5874 through 5876)

The free action is used to free up written tracks on thin devices.

Note: For Enginuity versions 5876 or lower, free actions are blocked for any device which is the source or target of a TimeFinder/Clone, Emulation, or Snap session.

For example, the following command starts freeing up space on thin devices in the device group `myDg`:

```
symdg -sid 234 -nop -v free -g myDg
```

The `symsg` and `symcg` commands are the same as `symdg` command using -sg *`SgName`* and the -cg *`CgName`*, respectively.

The following command starts freeing up space on thin device 1078 - 1081:

```
symdev -sid 234 -nop -v free devs 1078:1081
```

Restrictions for using free

- If a device group is specified, the action of the command is limited to the standard devices in the device group only.
- If the devices in a device group or storage group are bound to different pools, the free operation is allowed on the device group or storage group.
- Space freeing cannot be performed while a DATA device in the pool is draining.

Freeing all allocations or written tracks (HYPERMAX OS 5977 only)

On arrays running HYPERMAX OS 5977, you can free up allocations on thin devices using the `free -all` action. This action frees up allocations regardless of whether or not there is data written to the device, and is only available for arrays running HYPERMAX OS 5977. The device must be not mapped or not ready. The following commands support the `free -all` action:

For example, the following command starts freeing up space on thin devices in the device group `myDg`:

```
symdg -sid 234 -nop -v free -g myDg -all
```

The `symsg` and `symcg` commands are the same as `symdg` command using -sg *`SgName`* and the -cg *`CgName`*, respectively.

The following command starts freeing up space on thin device 1078 - 1081:

```
symdev -sid 234 -nop -v free devs 1078:1081 -all
```

The `free -all` command is blocked on any device participating in replication operations. The `symconfigure` command will not allow the following operations when a `free -all` operation is running in background on that device and it is in a ready state:

- add rdf mirror
- map dev
- convert dev

Stopping background operations on thin devices

Actions to allocate, free, or reclaim thin devices are performed asynchronously in the background and are started with an explicit start action. You can stop them, while in progress, using the `-stop` option. For example, when a `reclaim` operation is issued, the command returns as soon as Enginuity receives the request, and the operation takes place in background. Solutions Enabler displays these background tasks as the thin device status such as allocating, reclaiming, and so on.

Note: Stopping background operations is supported for HYPERMAX OS 5977 and all supported Enginuity versions.

For example, the following command stops allocation on thin devices in the device group `myDg`:

```
symdg -sid 234 -nop -v allocate -g myDg -stop
```

The `symsg` and `symcg` commands are the same as `symdg` command using -sg *`SgName`* and the -cg *`CgName`*, respectively.

The following command stops allocation on thin device 1078:

```
symdev -sid 234 -nop -v allocate devs 1078 -stop
```

Restrictions for stopping background operations

- If a device group is specified, the action of the command is limited to the standard devices in the device group.
- Devices must be in the allocating, reclaiming, or freeing state, otherwise the command will fail.

Draining and rebalancing a thin pool (Enginuity 5874 through 5876)

Solutions Enabler supports draining and rebalancing thin pools. The `symconfigure` command includes the following commands for these operations:

- `deactivate` — Disables one or more DATA devices without an automatic start of the device draining.
- `activate` — Enables one or more DATA devices without an automatic stop draining.
- `start drain` — Starts draining of one or more deactivated DATA devices.
- `stop drain` — Stops draining of one or more draining DATA devices.
- `start balancing` — Starts the rebalancing of devices in the pool.
- `stop balancing` — Stops the rebalancing of devices in the pool.

Draining rules

The following rules determine how many devices can be drained:

- The drain must not cause the enabled devices to end up with greater than 90% utilization in the pool. To calculate this, Solutions Enabler adds the total used tracks on the enabled devices and the total used tracks on the devices that will be drained and divides this sum by the total number of tracks on all the enabled devices. If the result is greater than 90% the drain request is blocked.
- No more than 20% of the enabled devices in a pool can be drained at any one time. This 20% limit includes the devices to be drained as well as the devices that will remain enabled in the pool.

Restrictions

The following restrictions apply to this feature:

- This feature is for thin pools only.
- All devices in the pools cannot be deactivated at the same time.
- The drain operation is not supported with any ongoing replication operation.

Deactivating devices in a thin pool

Use following command `symconfigure` command syntax to `deactivate` devices in a thin pool:

```
deactivate dev SymDevName [:SymDevName]in pool PoolName
type = thin;
```

For example, the following command deactivates device 1078 in pool `MyPool`:

```
symconfigure deactivate dev 1078 in pool MyPool type = thin
```

The `deactivate` command disables the DATA device without starting a drain. This operation puts the DATA device in either of following two states:

Deactivated — If the DATA device has any used tracks.

Disabled — If the DATA device has zero used tracks.

Note: The last DATA device of a pool in a thin tier cannot be deactivated, even if that pool has no bound devices.

DEACTIVATED state is determined by a combination of the INACTIVE state, the NOT_DRAINING state, and the presence of used tracks.

Reads and writes can be performed from previously allocated space on deactivated DATA devices. No new allocations can be performed on deactivated DATA devices.

The `deactivate` command is only supported for ENABLED DATA devices.

Starting the drain on DATA devices

Use the following `symconfigure` command syntax to start the drain on the deactivated DATA devices in a thin pool:

```
start drain on dev SymDevName [:SymDevName]in pool PoolName
type = thin;
```

For example, the following command starts the drain on device 1078 in pool `MyPool`:

```
symconfigure start drain on dev 1078 in pool MyPool type = thin
```

The drain progress can be viewed using the `symcfg` command. It can be viewed by listing the DATA devices in the array or using a detailed `show -pool`.

Activating devices in a thin pool

Use the following `symconfigure` command syntax to `activate` devices in a thin pool:

```
activate dev SymDevName [:SymDevName]in pool PoolName
type = thin;
```

For example, the following command activates device 1078 in pool `MyPool`:

```
symconfigure activate dev 1078 in pool MyPool type = thin
```

The `activate` command is essentially same as the `enable` command, however the `activate` command is not allowed if draining is in progress. After the `activate` command, the DATA device will go in the Enabled state.

The `activate` command is only supported for deactivated DATA devices with used tracks.

Stopping the drain on DATA devices

Use the following `symconfigure` command syntax to stop the drain on the DATA devices in a thin pool:

```
stop drain on dev SymDevName [:SymDevName]in pool PoolName
type = thin;
```

For example, the following command deactivates device 1078 in pool `MyPool`:

```
symconfigure stop drain on dev 1078 in pool MyPool type = thin
```

You can isssue the `stop drain` command on draining DATA devices to stop the drain operation. After the drain is stopped, the DATA devices are in the Deactivated state.

The `stop drain` command is only supported for draining DATA devices. Devices in any other state return an error.

Rebalancing example

A pool containing 32 DATA devices is 50% full. Sixteen new DATA devices are added to the pool. The following steps describe how to drain the 32 devices to the new 16 and rebalance the data load on the pool:

1. Deactivate 32 of the existing DATA devices. No new allocation will be allowed on these devices.
2. Perform the following actions until all 32 of the existing devices are processed:
   a. Start the drain on 4 of the 32 devices—these will be drained to the 16 new DATA devices.
   b. Monitor the drain operation using the `symcfg show -pool` until the 4 DATA devices drop a certain percentage in use.
   c. Stop the drain.
   d. Move on to the next set of 4 devices, performing steps a, b, and c for the rest of the original 32 DATA devices.

The result is the data is evenly distributed across the total 48 devices.

During this process, any new allocations on thin devices will write to the new 16 DATA devices only, because they are the only ones in an ENABLED state. You will need to factor in these writes while monitoring the drain on the 32 devices.

Starting pool rebalancing

When new devices are added to the pool, the allocation spread is no longer uniform and pool rebalancing is needed.

The `symconfigure` command provides the `start` action on a per pool basis.

On a `start` operation, Solutions Enabler will:

1. Check to see if a pool balancing operation is in progress for the specified pool.
2. If pool balancing is not in progress, Solutions Enabler will initiate pool balancing on the specified pool.

While write balancing is in progress, it is possible that data will be moved from deactivated DATA devices but never to deactivated DATA devices. The CLI output reports the pool state as Balancing. After the balancing operation completes, the pool state will not display Balancing. This pool state indicates if any balancing is in progress on a pool at any given time. All pool balancing operations must be initiated by the user.

Automated pool rebalancing can be invoked on any enabled pool. While pool rebalancing is in progress, all pool operations can still occur. Use the following command to start pool rebalancing on a thin pool:

```
start balancing on pool PoolName;
```

To start pool rebalancing on thin pool `XUN` on array ID `343`, enter:

```
symconfigure -sid 343 commit -file balance.cmd
```

where `balance.cmd` contains:

```
start balancing on pool XUN;
```

The `start balancing` operation is only supported on an enabled thin pool. To be enabled, a thin pool must have at least one DATA device in the enabled state.

For example, to check the status of the thin pool `XUN` on `343`, enter:

```
symcfg show -sid 343 -pool XUN -thin
```

```
Output similar to the following displays:
Symmetrix ID: 000190300343

Symmetrix ID                  : 000190300343
Pool Name                     : XUN
Pool Type                     : Thin
Dev Emulation                 : FBA
Dev Configuration             : 2-Way Mir
Pool State                    : Balancing
# of Devices in Pool          :   5
# of Enabled Devices in Pool  :   5

Enabled Devices(5):
  {
   ----------------------------------------------------------------
   Sym        Total      Used      Free  Full  Device    Session
   Dev       Tracks    Tracks    Tracks   (%)  State     Status
   ----------------------------------------------------------------
   04B6        8004      2000      6004    25  Enabled   N/A
   04B7        8004      2004      6000    25  Enabled   N/A
   04B8        8004       500      7504     6  Enabled   N/A
   04B9        8004       400      7604     5  Enabled   N/A
   04BA        8004      2000      6004    25  Enabled   N/A
           --------  --------  --------  ----
   Tracks     40020      6459     40020    16
```

Stopping pool rebalancing

Use the following command within a `symconfigure` command file to terminate a rebalancing operation on a thin pool:

```
stop balancing on pool PoolName;
```

The `stop balancing` operation is only supported on thin pools.

For example, when there is no balancing in progress, the output of the `symcfg show -pool` looks similar to the following:

```
Symmetrix ID: 000190300343

Symmetrix ID                  : 000190300343
Pool Name                     : XUN
Pool Type                     : Thin
```

```
Dev Emulation                 : FBA
Dev Configuration             : 2-Way Mir
Pool State                    : Enabled
# of Devices in Pool          :   5
# of Enabled Devices in Pool  :   5

Enabled Devices(5):
  {
   ---------------------------------------------------------------
   Sym        Total      Used      Free  Full  Device    Session
   Dev       Tracks    Tracks    Tracks   (%)  State     Status
   ---------------------------------------------------------------
   04B6        8004      1380      6624    17  Enabled   N/A
   04B7        8004      1400      6604    17  Enabled   N/A
   04B8        8004      1412      6592    18  Enabled   N/A
   04B9        8004      1312      6692    17  Enabled   N/A
   04BA        8004      1400      6604    17  Enabled   N/A
           --------  --------  --------  ----
   Tracks     40020      6459     40020    16
```

Automated pool rebalancing

Whenever writes occur on a thin device, the actual data is written to the DATA devices in the thin pool to which the thin device is bound. The array does spread out the I/O equally on all the DATA devices in the thin pool, therefore, at any given point in time the used capacity of each DATA device is uniform across the pool.

Virtual Pool (VP) compression (Enginuity 5876 only)

For Enginuity 5876 Q42012SR, the VP compression feature for thin pools and thin devices is supported. Data can be compressed and uncompressed manually or can be compressed by FAST as part of its processing.

Note: For arrays running HYPERMAX OS 5977, VP compression is not supported, and setting the VP compression attribute and compression start/stop commands is blocked.

Data that is compressed can be read and written. A pool must be enabled for compression to allow for data compression. This is done by setting the compression capability on a VP Pool to enabled. When compression is no longer desired for allocations in a pool, the compression capability on a pool can be disabled.

Once a pool is enabled for compression, a background process runs to reserve storage in the pool for temporarily uncompressing data. No compressions of allocations in this pool can occur until this process completes. Disabling the compression capability on a pool does not automatically cause compressed allocations to be uncompressed. This must be done manually.

When you compress a device, only allocations that have been written are compressed. Any allocations that were created during the bind or allocate process that have not been written are reclaimed during the compression process. In order to allow the compression of persistent allocations, which implies the possible reclaiming of allocations that have not been written, the persistent indicator on the allocations must be manually removed. Once the compression request is accepted, the thin device will have a background task associated with it that performs the compression. While this task is running, no other background task, such as allocate or reclaim, can be run against the thin device.

Compression is set with pool attributes as explained in "Setting pool attributes (Enginuity 5773 through 5876)" on page 283.

The `symconfigure` command is used to enable and disable VP Compression for a VP pool on the array.

Note: The pool must be a VP pool. Compression cannot be set on a pool of encapsulated data devices and Snap VP target devices.

Once a pool has been enabled for compression, one of the following CLI commands can be used to start/stop VP compression or VP uncompression for allocations on a thin device:

```
symdev -sid SymmID [-noprompt]
   -devs [SymDevStart:SymDevEnd | SymDevName
                [,SymDevStart:SymDevEnd | SymDevName...]]
    compress [-stop]
    uncompress [-stop]
```

A file can be used to specify the list of devices to be acted upon. The file must contain one SymDevName per line. When using a file the syntax is `-file <FileName>`.

Compressing or uncompressing a thin device occurs as a background task. Use the `symcfg show -detail` or `symcfg list -devs` commands to view a percentage of tracks compressed as displayed in the `Compressed Size/Ratio Tracks %` column.

VP compression rules

The following rules apply for VP compression processing:

- Devices in a thin pool that are enabled for compression become uncompressed when tracks are written to or changed as a result of another action, therefore, compressing devices is not permanent. You must continually compress a device if the tracks become uncompressed.
- If all the targeted devices are already in the state for the specified action (that is, all are compressing and a compress start was issued), then the command fails the request with a SYMAPI_C_ALREADY_IN_STATE status.
- If some of the targeted devices are in the desired state (that is, compressing on a compress start action) and the remaining devices do not have a conflicting state (that is, in a uncompressing state when the action is compress start), then the command processes the request and skips the devices that are already in the desired state.
- If any of the targeted devices are in a conflicting state relative to the command (that is, some devices are in a compressing state and the action is uncompress start), then the command fails the request with the SYMAPI_C_INVALID_STATE_FOR_ACTION status.
- If compression has not been enabled for any pool in which a device has allocations, the CLI does not fail a compress start action, but compression does not occur for those allocations.
- Any non-thin device is skipped.
- Any allocations that are not written are reclaimed during compression if the tracks are not persistently allocated. This implies that after an uncompression, the allocated tracks do not match the original allocated track count for the TDEV. By making the

allocations persistent, the allocated tracks are maintained and no compression takes place. If allocations are persistent and a reclaim of space is desired, then the persistent attribute for existing allocations may be cleared, allowing compression and reclamation to occur.

Verifying the pool and device states

The `symcfg verify` command allows you to verify the states of DATA devices and thin devices. It also allows you to determine if the pool is in a valid pool state.

Note: Verifying pool and device states is supported for HYPERMAX OS 5977 and all supported Enginuity versions.

To verify is a pool is enabled, disabled, the standard verification options are provided, such as blocking until the pool is in the desired state, and polling at a given rate. To verify a pool state, use the following form:

```
symcfg -sid SymmID [-i Interval] [-c Count]
        <-pool PoolName |-devs <SymDevStart:SymDevEnd |
           SymDevName
      [,<SymDevStart:SymDevEnd | SymDevName>...]>>

   verify -datadev
   <-draining | -drainwait | -disabled | -enabled |
   -deactivated | -nonpooled | -balancing>

symcfg -sid SymmID [-i Interval] [-c Count]
        -pool PoolName

   verify -poolstate
   <-enabled | -disabled | -balancing>
```

Monitoring thin pools

The `symcfg monitor` command allows you to monitor a thin pools usage and take action if the `-percent` usage reaches a certain threshold.

Note: Monitoring thin pools is supported for HYPERMAX OS 5977 and all supported Enginuity versions.

To monitor a thin pool's usage, use the `symcfg monitor` command:

```
symcfg [-sid SymmID] [-i Interval] [-c Count] [-offline]
         [-percent <1-100> -action ScriptFile [-norepeat]]
         [-snap | -rdfa_dse | -thin] [-pool PoolName]
         [-mb] [-gb]
       monitor
```

Where:

- `-action` — Selects a script that should be run when the specified percent value is encountered. The full pathname to the action script must be specified. The first argument passed to the script is automatically set to the percent value.
- `-c` — Indicates the number (count) of times to display or to acquire an exclusive lock on the array host database, the local array, and the remote arrays.

- `-i` — Specifies the repeat interval in seconds to display or to acquire an exclusive lock on the array host database, the local , and the remote arrays. The default interval is 10 seconds. The minimum interval is 5 seconds.
- `monitor` — Checks the total percent full of the devices currently configured in the pool and can optionally execute a script file if a specified percentage is encountered.
- `-norepeat` — Specifies that the action script should only be run once if the threshold has been met. Used with the action script option on the `monitor` command.
- `-percent`— Causes the action script to be executed when the percent full argument is encountered.
- `-pool` — Specifies a thin pool.
- `-thin` — Specifies that the pool type is a thin pool.

Viewing thin pools

When you add the `-tdev` option, thin devices display with all pools in which they have allocated tracks. The pool allocated tracks are listed per pool in which the device has allocations. The pool written tracks are only reported for the bound pool and represents all tracks that have been written to the thin device for any allocation in any pool. The compression is reported by track size (physical size) and the compressed ratio percentage.

Note: Viewing thin pools is supported for HYPERMAX OS 5977 and all supported Enginuity versions.

The following is an example command and sample output for the `symcfg list -tdev -detail` command:

```
symcfg list -tdev -detail -sid 086
```

```
  . . .

Enabled Capacity (Tracks) :  157681440
Bound   Capacity (Tracks) :    4257765

                    S Y M M E T R I X   T H I N   D E V I C E S
--------------------------------------------------------------------------------
   ----
                                   Pool        Pool          Total      Compressed
       Bound      Flags      Total  Subs      Allocated       Written     Size/Ratio
Sym   Pool Name    ESPT      Tracks   (%)     Tracks (%)     Tracks (%)     Tracks (%)
----- ------------ ----- ---------- ----- ---------- --- ---------- --- ----------
   ---
00001 -           F..B       26220     0          0   0          0   0          0   0
     DG2_FBA_PL   -.--          -     -      26220 100          -   -      26220   0
00002 -           F..B       26220     0          0   0          0   0          0   0
  . . .

Total                  ---------- ----- ---------- --- ---------- --- ---------- ---
Tracks                     4257765     3      26220   2          0   0      26220   0

Legend:
Flags:  (E)mulation : A = AS400, F = FBA, 8 = CKD3380, 9 = CKD3390
         (S)hared Tracks : S = Shared Tracks Present, . = No Shared Tracks
         (P)ersistent Allocs : A = All, S = Some, . = None
         S(T)atus    : B = Bound, I = Binding, U = Unbinding, A = Allocating,
```

```
                    D = Deallocating, R = Reclaiming, C = Compressing,
                    N = Uncompressing, F = FreeingAll, . = Unbound
```

For arrays running HYPERMAX OS 5977, the `symcfg list -tdev` display reports a dash (-) in the Bound Pool Name column for all TDEV devices. The `symcfg list -tdev detail` display also reports a dash (-) in the Bound Pool Name column for all TDEV devices. but only for the first (summary) line for each TDEV. If the TDEV has allocations in other VP Pools, the pool names and allocation details for those pools will be displayed on subsequent lines.

The following example shows the command and output for the `symcfg list -pool -all` command:

```
symcfg list -pool -all -sid 397

Symmetrix ID: 000195700397

                   S Y M M E T R I X   P O O L S
-------------------------------------------------------------------------------
Pool         Flags  Dev              Usable       Free       Used Full Comp
Name         PTECSL Config           Tracks     Tracks     Tracks  (%)  (%)
------------ ------ ------------- ---------- ---------- ---------- ---- ---
DEFAULT_POOL SSFDEI 2-Way Mir        600000     600000          0    0    0
  . . .
Pool1        TSFDEI 2-Way Mir      12923748   12945324     105408    0    0
  . . .
Mig_src2     TSFEEI RAID-6(6+2)       19224      16824       2400   12   29
Mig_trg2     TSFDEI 2-Way Mir         19224      19188         36    0    0

Total                            ---------- ---------- ---------- ---- ----
Tracks                             15638331   15616299     149016    0    2

Legend:
  (P)ool Type:
    S = Snap,  R = Rdfa DSE  T = Thin
  (T)echnology:
    S = SATA, F = Fibre Channel, E = Enterprise Flash Drive, M = Mixed, - = N/A
  Dev (E)mulation:
    F = FBA, A = AS400, 8 = CKD3380, 9 = CKD3390, - = N/A
  (C)ompression:
    E = Enabled, D = Disabled, N = Enabling, S = Disabling, - = N/A
  (S)tate:
    E = Enabled, D = Disabled, B = Balancing
  Disk (L)ocation:
    I = Internal, X = External, M = Mixed, - = N/A
```

The following example shows the output when the pool does contain allocations from devices that are bound to a different pool:

For arrays running HYPERMAX OS 5977, the `symcfg show -pool -thin -detail` and `symcfg list -pool -v -thin -detail` command displays are changed as follows:

- Pool Bound Thin Devices section displays as No Thin Devices Bound to Device Pool
- Other-Pool Bound Thin Devices section displays As Other Thin Devices with Allocations in this Pool.

```
symcfg show -pool Mig_trg2 -thin -sid 432 -all -detail

Symmetrix ID: 000195700432

Symmetrix ID                     : 000194900432
Pool Name                        : Mig_trg2
Pool Type                        : Thin
```

```
Disk Location                    : Internal
Technology                       : SATA
Dev Emulation                    : FBA
Dev Configuration                : 2-Way Mir
Pool State                       : Enabled
Compression State                : Enabled
# of Devices in Pool             :  10
# of Enabled Devices in Pool     :  10
# of Usable Tracks in Pool       : 99000
# of Allocated Tracks in Pool    : 14081
# of Thin Device Tracks          : 11549
# of DSE Tracks                  : 2298
# of Local Replication Tracks    : 234
# of Tracks saved by compression : 0
# of Shared Tracks in Pool       : 0
Pool Utilization (%)             :   3
Pool Compression Ratio (%)       :   0
Max. Subscription Percent        : N/A
Rebalance Variance               : N/A
Max devs per rebalance scan      : N/A
Pool Reserved Capacity           : N/A

Enabled Devices(6):
  {
   ----------------------------------------------------------
   Sym        Usable      Alloc       Free Full FLG Device
   Dev        Tracks     Tracks     Tracks  (%)  S  State
   ----------------------------------------------------------
   0811         3204        456       2748   14  .  Enabled
   0812         3204        456       2748   14  .  Enabled
   0813         3204        444       2760   14  .  Enabled
   082A         3204        444       2760   14  .  Enabled
   082B         3204        444       2760   14  .  Enabled
   082C         3204        444       2760   14  .  Enabled
          ---------- ---------- ---------- ----
   Tracks      19224       2688      16536   14
  }

Pool Bound Thin Devices(3):
{
   ---------------------------------------------------------------------
                      Pool          Pool          Total        Compressed
   Sym   FLG       Total Sub      Allocated      Written       Size/Ratio
   Dev    T       Tracks (%)     Tracks (%)     Tracks (%)     Tracks (%)
   ---------------------------------------------------------------------
   0816   B         4800  25        144   3          0   0        144   0
   082F   B         4800  25       1440  30          0   0        720  50
   086E   B          750   4        372  50          0   0        252  32
             ---------- --- ---------- --- ---------- --- ---------- ---
   Tracks          10350  54       1956  19          0   0       1116  42
  }

Other-Pool Bound Thin Devices(2):
{
   ----------------------------------------------------------
                                      Pool        Compressed
          Bound           Total      Allocated     Size/Ratio
   Sym  Pool Name        Tracks     Tracks (%)     Tracks (%)
   ----------------------------------------------------------
   0826 Mig_src2           6000        360   6        360   0
   092C HR_Archive         3000        372  12        252  32
                     ---------- ---------- --- ---------- ---
   Tracks                  9000        732   8        612  16
  }

Legend:
```

```
Enabled devices FLG:
  (S)hared Tracks : X = Shared Tracks , . = No Shared Tracks
Bound devices FLG:
  S(T)atus    : B = Bound, I = Binding, U = Unbinding, A = Allocating,
                D = Deallocating, R = Reclaiming, C = Compressing,
                N = Uncompressing, F = FreeingAll, . = Unbound,
```

CHAPTER 9
Federated Tiered Storage

This chapter describes Federated Tiered Storage (FTS) concepts and explains how to configure and manage FTS using the SYMCLI.

- What is FTS? 302
- Using eDisks 303
- Geometry of encapsulated devices (Enginuity 5876) 304
- Creating an external disk group (Enginuity 5876 or lower) 306
- Deleting a disk group (Enginuity 5876 or lower) 306
- Adding an eDisk 307

What is FTS?

Federated Tiered Storage (FTS) gives you the ability to attach external storage to VMAX arrays. Attaching external storage allows you to use physical disk space on existing arrays while gaining access to VMAX features such as local replication, remote replication, storage tiering, data management, and data migration. In addition, FTS simplifies the management of federated multi-vendor or EMC storage arrays.

FTS for HYPERMAX OS 5977

For VMAX arrays with HYPERMAX OS 5977, FTS support is limited to virtual provisioning encapsulation of Data Domain eDisks for EMC ProtectPoint.

ProtectPoint integrates backup and restore operations by allowing you to backup directly from VMAX to Data Domain platforms. By integrating VMAX and Data Domain storage, ProtectPoint reduces cost and complexity by eliminating traditional backup applications while still providing the benefits of native backups. With ProtectPoint you can:

- Achieve faster, more frequent backups to meet stringent Service Level Objectives
- Instantly access application backups from Data Domain for simple granular recovery
- Recover faster from native full backups
- Virtually eliminate backup impact on application servers
- Eliminate the need for a dedicated backup server

Note: The *Solutions Enabler TimeFinder Family CLI User Guide* provides details on using the SYMCLI to perform ProtectPoint operations.

Support is provided for one external disk group and one associated pool to store external devices. The disk group and pool are created automatically during the VMAX array pre-configuration process and all encapsulated virtually-provisioned devices are added during this process.

FTS for Enginuity 5876

Solutions Enabler V8.0.1 with Enginuity 5876 supports FTS on VMAX 10K, 20K, and 40K platforms. For detailed information about FTS configuration and operating modes, refer to the *EMC Symmetrix VMAX Family with Enginuity (10K, 20K, 40K) Product Guide.*

For VMAX arrays running Enginuity 5876, you can perform the following disk group management operations:

- Create and name an empty external disk group
- Delete an empty external disk group

Using eDisks

When you attach external storage to a VMAX array, FTS virtualizes an external array's SCSI logical units as VMAX disks called eDisks. eDisks have two modes of operation:

- Encapsulation — Allows you to preserve existing data on external arrays and access it through VMAX devices. These devices are called encapsulated devices.
- External Provisioning — Allows you to use external storage as raw capacity for new VMAX devices. These devices are called externally provisioned devices. Existing data on the external devices is deleted when they are externally provisioned.

The following restrictions apply to eDisks:

- Can only be unprotected devices. The RAID protection scheme of eDisks is dependent on the external array.
- Cannot be AS400, CKD, or gatekeeper devices.
- Cannot be used as VAULT, SFS, or ACLX devices.

Encapsulation

Encapsulation has two modes of operation:

- Encapsulation for disk group provisioning (DP encapsulation) — The eDisk is encapsulated and exported from the VMAX array as disk group provisioned devices.
- Encapsulation for virtual provisioning (VP encapsulation) — The eDisk is encapsulated and exported from the VMAX array as thin devices.

In either case, Enginuity automatically creates the necessary VMAX devices. If the eDisk is larger than the maximum VMAX device size or the configured minimum auto meta size, Enginuity creates multiple VMAX devices to account for the full size of the eDisk. These VMAX devices are concatenated into a single concatenated meta device to allow access to the complete volume of data available from the eDisk.

External provisioning (Enginuity 5876 only)

When you virtualize an eDisk for external provisioning, you can then create VMAX devices from the external disk group and present the storage to users. You can also use this storage to create a new FAST VP tier.

The FTS tier can be associated with a technology type. The technology associated with the FTS tier indicates to the FAST VP controller the expected performance from the tier. This feature enables you to place the FTS tier at the right location for the expected performance of the external tier.

See "Fully Automated Storage Tiering (Enginuity 5874 to 5876)" on page 166 for additional details.

Geometry of encapsulated devices (Enginuity 5876)

Enginuity 5876 builds VMAX devices based on the VMAX cylinder size (fifteen 64 K tracks), so the size of devices will not always match the raw capacity of the eDisk. If the size does not match, Enginuity sets a custom geometry on the encapsulated device. This custom geometry is reported as user defined geometry in Solutions Enabler.

For created meta devices, Enginuity defines the geometry on the meta head, and only the last member can have a size that spans beyond the raw capacity of the eDisk.

Geometry limited devices

Encapsulated devices that have a VMAX cylinder size larger than the reported user-defined geometry size are considered geometry limited. The following restrictions apply to geometry-limited devices:

- Does not support TimeFinder/Snap or TimeFinder/Mirror. Can only be used as source devices for TimeFinder/Clone operations. The rules for operations to larger target devices apply. See the *Solutions Enabler TimeFinder Family CLI User Guide* for details about copying data from a source device to a larger target device.

  If the devices are meta devices, the following requirements must be met:

  – The number of meta members on the source device and the target device must be the same.

  – For each of the source device and target device meta members, the VMAX cylinder size must be the same.

- Can only be used as R1 devices for SRDF migration operations. The rules for operations to larger R2 devices apply. See the *EMC Solutions Enabler SRDF Family CLI User Guide* for details about copying data from an R1 device to a larger R2 device. Does not support SRDF/AR.

  If the devices are meta devices, the following requirements must be met:

  – The number of meta members on the device that contains the R1 and the device that contains the R2 must be the same.

  – For each of the meta members on the device that contains the R1 and the device that contains the R2, the VMAX cylinder size must be the same.

- Cannot be expanded, dissolved, converted, or used as the target of a VLUN migration.
- Thin devices cannot be unbound or rebound from their pool, and their space cannot be reclaimed.
- Thin pools created during encapsulation cannot have data devices added to them unless the addition is the result of encapsulating external storage.
- Thin pools cannot be added to tiers.

Encapsulation examples

This section provides examples of geometry-limited encapsulation and encapsulation that is not geometry limited.

Geometry-limited encapsulation

Assume that you virtualize an eDisk and encapsulate the data for a 22 GB external device with the automatic meta member size on the VMAX array configured to 9 GB. The following actions occur:

1. Enginuity creates three 9 GB VMAX devices and associates each one with an unprotected, virtual RAID group.
2. Enginuity concatenates the VMAX devices into a single concatenated meta device.
3. Enginuity sets the user defined geometry to reflect the encapsulated 22 GB capacity.

This device is considered geometry limited because the VMAX device size reported on the meta head (27 GB) is larger than the user defined geometry size of 22 GB.

Encapsulation not geometry limited (meta rules enforced)

Assume that you virtualize an eDisk and encapsulate the data for a 27 GB external device with the automatic meta member size on the VMAX array configured to 9 GB. The following actions occur:

1. Enginuity creates three 9 GB VMAX devices and associates each one with an unprotected. virtual RAID group.
2. Enginuity concatenates the VMAX devices into a single concatenated meta device.
3. Enginuity sets the user defined geometry to reflect the encapsulated 27 GB capacity.

This device is not considered geometry limited because the size reported on the meta head (27 GB) and the user defined geometry (including the size in bytes) match exactly.

Encapsulation that is not geometry limited (exact device size enforced)

Assume that you virtualize an eDisk and encapsulate the data for a 22 GB external device, and you override the automatic meta member size configured on the VMAX array by specifying the exact device size. The following actions occur:

1. Enginuity creates two 9 GB VMAX devices and one 4 GB VMAX device and associated each one with an unprotected, virtual RAID group.
2. Enginuity concatenates them into a single concatenated meta device.
3. Enginuity sets the user defined geometry to reflect the exact encapsulated 22 GB capacity.

This device is not considered geometry limited because the VMAX device size reported on the meta head (22 GB) and the user defined geometry (including the size in bytes) match exactly.

Creating an external disk group (Enginuity 5876 or lower)

Note: This command is blocked for arrays running HYPERMAX OS 5977 or higher.

To use FTS, you must set up an external disk group using the `symconfigure` command. Use the following syntax to create an empty disk group:

```
create disk_group DskGrpName disk_location = external;
```

The disk group name serves as an additional identification mechanism and does not replace disk group numbers.

For example, to create a disk group for a storage pool and give it the name `hr_sg_pool`, enter the following from the command line:

```
symconfigure -sid 2012 -cmd "create disk_group hr_disk_group
 disk_location = external;" commit
```

Restrictions

The following restrictions apply when creating disk groups.

- The maximum number of external disk groups that can be created on an array is 512.
- A disk group cannot be created and deleted in the same session.
- The specified disk group name must be unique; the check for uniqueness is case insensitive.
- The specified disk group name cannot exceed 32 characters. Only alphanumeric characters, hyphens "-" and underscores "_" are allowed. The name cannot start with a hyphen "-" or underscore "_".
- The disk group name given cannot have the same format as the default disk group names, for example DISK_GROUP_001.

Deleting a disk group (Enginuity 5876 or lower)

Use the following form to delete a disk group by disk group number or by disk group name:

```
delete disk_group <DskGrpNum | name:DskGrpName>;
```

For example, to delete a disk group with the name `hr_disk_group`, enter the following from the command line:

```
symconfigure -sid 2012 -cmd "delete disk_group name:hr_disk_group;" commit
```

Restrictions

The following restrictions apply when deleting disk groups:

- The disk group must be empty.
- The disk group name or number must exist.
- Only external disk groups can be deleted.

Physical disks and disk groups have some rules that need to be observed by this new feature, as follows:

- Managing external disk groups requires Enginuity 5876 or HYPERMAX OS 5977.
- An external disk group can only contain external disks.
- Every external disk in the array must belong to an external disk group.

Adding an eDisk

When you add an eDisk to a VMAX array, you must add it to an external disk group. You can use the `symsan` command to obtain the WWN of the external LUN that you plan to add. Refer to the *EMC Solutions Enabler SYMCLI Command Reference* for the `symsan` manpage.

The following restrictions apply when adding an eDisk:

- eDisks can only be associated with one external spindle at a time.
- The disk group for the eDisk must already exist (or be created first in the same command).
- With virtual provisioning, either an empty pool or an existing pool composed of external data devices must be provided.

To add an eDisk to an external disk group, create and commit a `symconfigure` command file with the following syntax:

```
add external_disk wwn=wwn to
   disk_group=<DskGrpNum | name:DskGrpName>,
   encapsulate_data=<NO | YES [pool = PoolName]>
   [meta_member_size = n [MB | GB | CYL]]
   [member_size_equal=YES | NO] [dir = director_num];
```

For example, to add an eDisk for external provisioning to disk group `exchange_disks1`, create and commit a command file that contains the following command:

```
add external_disk wwn=60000970000194900306533030314341 to disk_group=exchange_disks1,
encapsulate_data=NO;
```

For example, to add an eDisk to disk group `exchange_disks2` and encapsulate the data using DP encapsulation, create and commit a command file that contains the following command:

```
add external_disk wwn=60000970000194900306533030314345 to disk_group=exchange_disks2,
encapsulate_data=YES;
```

For example, to add an eDisk to disk group `exchange_disks2` and encapsulate the data using VP encapsulation (with a thin pool named `eng_pool`), create and commit a command file that contains the following command:

```
add external_disk wwn=60000970000194900306533030314345 to disk_group=exchange_disks2,
encapsulate_data=YES, pool=eng_pool;
```

Restrictions

For arrays running HYPERMAX OS 5977, the following restrictions apply:

- The options `to disk_group` and `pool` are not allowed.

- You cannot set the `encapsulate_data` option to NO; the only valid value is YES.
- You cannot specify either the `meta_member_size` or `member_size_equal` option.

If any of the above options are attempted, the command will be rejected.

For arrays running Enginuity 5773 through 5876, note the following behavior and requirements when constructing your command file:

- If the `encapsulate_data` option is set to NO, any data that is currently on the external device is deleted.
- If you are using virtual provisioning, you must provide the name of an existing thin pool.
- You can override the auto meta member size configured on the VMAX array by specifying the meta member size (`meta_member_size = n [MB | GB | CYL]`).
- If you do not specify that all of the created VMAX devices be the same size (`member_size_equal=YES`), the meta tail will be smaller than the other devices in the meta.

Removing an eDisk

The following rules apply when removing the eDisk:

- If it was added with encapsulation, the devices on the eDisk must be unmapped and must not be part of a migration, local replication, remote replication, or ORS session. RAID groups and any disk group provisioned devices or DATA devices created during the encapsulation are removed. Thin devices are unbound but they are not removed.
- If it was added for external provisioning, the VMAX devices on the eDisk must be deleted before it can be removed.

To remove an eDisk, create and commit a command file that contains the following syntax:

```
remove external_disk <wwn=<wwn> | spid=<SpindleID>>
   [force_remove=<YES | NO>];
```

Where:

`force_remove` — This option allows forcing the removal of an external encapsulated disk when data is still in the VMAX cache. Specifically this can occur when the device has iVTOC and write pending tracks.

To remove an eDisk by specifying the eDisks WWN, create and commit a command file that contains a command similar to the following:

Example

```
remove external_disk wwn=60000970000184700306533030314345;
```

To remove an eDisk by using the specifying the spindle ID, create and commit a command file that contains a command similar to the following:

```
remove external_disk spid=2256;
```

Displaying external disk information

The following example shows the information displayed for external disks:

```
symdisk show 07a:c0 -sid 432

Symmetrix ID                  : 000194900432
  Director                    : DF-7A
  Interface                   : C
  Target ID                   : 0
  Spindle ID                  : 2D0
    . . .
  Hypers (189):
  {
  #   Vol   Emulation        Dev   Type          Mir Mbr Sts Cap(MB)
  --- ----- ---------------- ----- ------------- --- --- --- --------
    1     1 VAULT_DEVICE     00000 Data           1  N/A N/A     5200
    2     2 FBA              001A9 RAID-5         1   1  WD     34179
   43    43 CKD-3390         000C0 RAID-1         1   1  NR       836
  189   189 FBA              00464 RAID-1         1   2  RW     31203
  }
```

Solutions Enabler FTS support (Enginuity 5876 only)

FTS supports the Solutions Enabler features listed in Table 20:

Table 20 **FTS support across Solutions Enabler features (Enginuity 5876)**

Feature	Support details
FAST VP	Supports VP pools of externally provisioned DATA devices. Solutions Enabler allows you to specify the technology type of the FTS tier. In addition, FAST VP supports up to four tiers when one of the four tiers is the external FTS tier.
Open Replicator	• Full support for externally provisioned device for ORS, RP, and FLM operations. • Support for encapsulated devices that are not geometry limited for only ORS and RP operations. • Encapsulated devices are not supported for FLM operations. • Encapsulated devices that are geometry limited are not supported for ORS pull operations.
SRDF	• All SRDF operations are supported for externally provisioned devices with any RDF personality (R1, R2, etc.). • All SRDF operations are supported for encapsulated devices that are not geometry limited with any RDF personality, (R1, R2, etc.). • Encapsulated devices that are geometry limited are supported for SRDF migration operations only. Support is limited to R1 devices. The rules of operation (R2 larger than R1) apply.
SRDF/SAR	The `symreplicate` CLI supports both encapsulated and externally provisioned devices, as follows: • Supports externally provisioned devices. • Supports encapsulated devices that are not geometry limited. • Does not support encapsulated devices that are geometry limited for any SAR operation. The `symrecover` CLI supports both encapsulated and externally provisioned devices, as follows: • Supports externally provisioned devices. • Supports encapsulated devices that are not geometry limited. • Does not support encapsulated devices that are geometry limited.
SRDF/Star	• Supports externally provisioned devices. • Supports encapsulated devices that are not geometry limited. • Does not support encapsulated devices that are geometry limited for any Star operation.

Table 20 **FTS support across Solutions Enabler features (Enginuity 5876)**

Feature	Support details
TimeFinder	• Supports externally provisioned devices as source and target devices for TimeFinder/Snap, Clone, and Mirror (TF/Clone Emulation) operations. • Supports encapsulated devices that are not geometry limited as source and target devices for TimeFinder/Clone, and TimeFinder/Mirror (TF/Clone Emulation) operations. • Supports encapsulated devices that are geometry limited only for TimeFinder/Clone operations. These devices can be used as a source devices and the rules for operations to larger target devices apply.
VLUN migration	Supports both encapsulated and externally provisioned devices, however encapsulated devices cannot be used as targets of a VLUN migration and cannot be migrated using the configured space option.

CHAPTER 10
Masking Devices with Auto-Provisioning Groups

This chapter explains how to confine host access to array devices using auto-provisioning groups and the SYMCLI `symaccess` command.

- Auto-provisioning groups 312
- Creating an auto-provisioning session 314
- Managing masking views 319
- Managing storage groups 321
- Managing port groups 323
- Managing initiator groups 326
- Displaying auto-provisioning group information 332

Auto-provisioning groups

Auto-provisioning groups allow storage administrators to create groups of host initiators, front-end ports, and logical devices. These groups are then associated to form a masking view, from which all controls are managed. This feature reduces the number of commands needed for masking devices, and allows for easy management of the masking view.

Storage provisioning with `symaccess` allows you to create a group of devices, a group of director ports, a group of host initiators, and with one command, associate them in what is called a *masking view*. Once a masking view exists, devices, ports, and initiators can be easily added or removed from their respective groups.

The `symaccess` command also contains some features similar to those found in the `symmaskdb` command (a login history command and initiator attributes). However, two `symmask` operations continue to be supported for arrays running Enginuity 5874 and higher: the backup and restore of the masking database, and listing of the attached Host Bus Adapters.

Arrays running HYPERMAX OS 5977 or higher do not support auto-provisioning for iSCSI. This includes the iSCSI front-end director emulation and attached ports, iSCSI initiators, and CHAP security assignment to iSCSI ports.

Auto-provisioning for iSCSI is supported for arrays running Enginuity 5874 through 5876.

Note: Auto-provisioning groups is not supported on DMX arrays running Enginuity 5773 or lower. The `symmask` and `symmaskdb` commands should be used to mask devices in DMX arrays.

Provisioning limits

Provisioning limits for the VMAX arrays are shown in Table 21 and Table 22.

Table 21 Provisioning limits for arrays running HYPERMAX OS 5977 or higher

Provisioning object	Configuration limitations
Devices	Maximum 16K per each director Maximum 4K per each storage group
Initiator group	Maximum 64 initiator addresses (or 64 child IG names) per group
Storage group	Maximum 16K storage groups per array Maximum 64 child storage groups per each parent storage group Maximum 4K storage groups with host I/O limits defined
Port group	Maximum 16K port groups per array Maximum 32 ports in a port group
Masking view	Maximum 16K masking views per array
LUN addresses	Maximum 4K LUN addresses per director port

Table 22 Provisioning limits for VMAX arrays running Enginuity 5874 through 5876

Provisioning object	Configuration limitations
Devices	Maximum 4K per each director port Maximum 4K per each storage group
Initiator group	Maximum 32 initiator addresses (or 64 child IG names) per group
Storage group	Maximum 4K storage groups per array Maximum 32 child storage groups per each parent storage group Maximum 2K storage groups with host I/O limits defined
Port group	Maximum 8K storage groups per array
Masking view	Maximum 8K masking views per array
LUN addresses	Maximum 4K LUN addresses per director port

Creating a masking view

The steps for creating a masking view are:

1. Create a storage group (one or more devices).
2. Create a port group (one or more director/port combinations).
3. Create an initiator group (one or more host WWNs or iSCSIs).
4. Create a masking view containing the storage group, port group, and initiator group.

When a masking view is created, the devices are automatically masked and mapped.

Figure 8 shows a masking view.

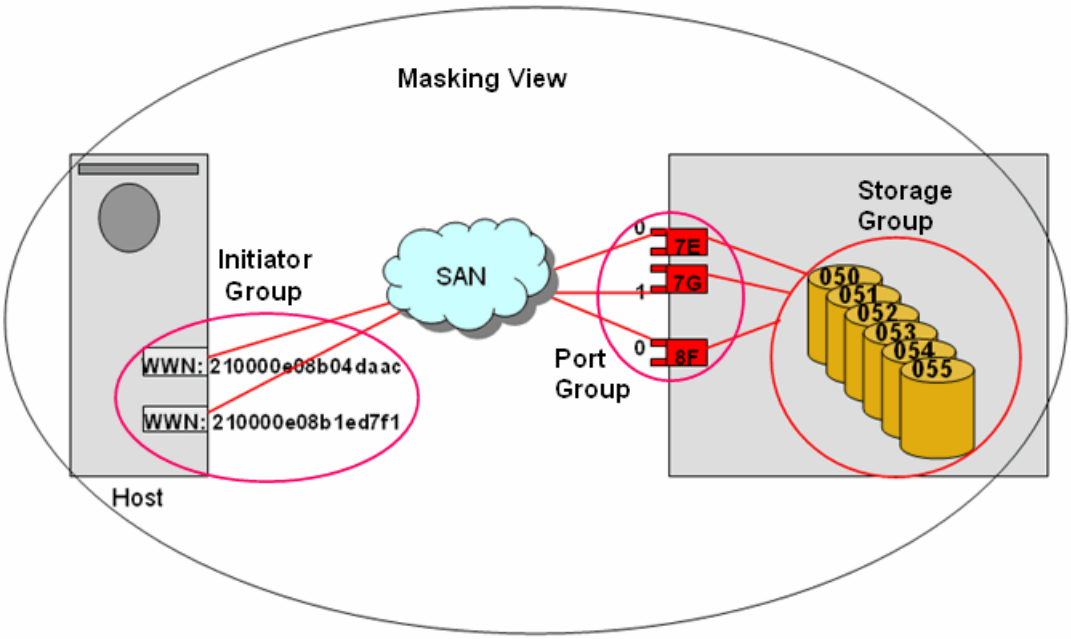


Figure 8 Masking view overview

After the masking view is created, any objects (devices, ports, initiators) added to a group automatically become part of the masking view.

Session rollback

When an auto-provisioning session fails on arrays running Enginuity 5876 or higher, the system automatically rolls back the ACLX database to the state in was in prior to initiating the session. This rollback feature recovers the database and releases the session lock automatically. The audit log contains any messages relating to the rollback.

Performing auto-provisioning operations to create views, or add devices or ACLX-enabled ports to existing views will continue to implicitly map devices to ports, but if the masking view update fails these devices will be unmapped when the session is rolled back.

Note: For arrays running HYPERMAX OS 5977 or higher, explicit mapping and unmapping of ACLX-enabled ports using the `symconfigure` command is not supported. To make the ACLX device visible or to remove visibility from the ACLX device, you must modify the new SHOW_ACLX_DEVICE attribute on front end ports. By default, ACLX-enabled ports have the SHOW_ACLX_DEVICE attribute disabled, making the ACLX device not visible to hosts zoned to the port. One ACLX-enabled port is pre-configured with the SHOW_ACLX_DEVICE attribute enabled. During device remove, port remove, or masking view delete operations, the `-unmap` option is ignored for HYPERMAX OS 5977 or higher.

Storage groups containing CKD devices must already be mapped, and must use the optional flag `-ckd`. Storage groups containing Celerra devices can be masked (and mapped) by using the `-celerra` option. Storage groups containing devices tagged for RecoverPoint can be masked and mapped by using the `-rp` option.

Creating an auto-provisioning session

This section details the steps for creating an auto-provisioning session.

Discovering host HBAs

During the initial setup, an administrator runs `symaccess discover` on the controlling host to search the environment for array devices on each HBA by using the following command:

```
symaccess discover hba
```

When the `symaccess discover` command finds a host HBA, it reads the login history table and performs the following:

1. Creates an ASCII alias and writes it to the login history table.

   Note: There is a `-rename` option that can be used with this command to force the discovered hostname/HBA name (or IP address) to be written to the login history table and the initiator group.

2. Prints the initiator identifier (WWN/iSCSI) of the HBAs that are connected to the masked channel and the array.

The `symaccess discover` command sends information about this connection back to its host system. The `discover` command is the primary mechanism by which hosts other than the control station can learn about their paths to the array. This information can be displayed using the `symaccess list hba` command.

Note: An iSCSI initiator cannot log in to the array until it belongs to a masking view that includes that specific port on the array.

Creating groups and views

The `symaccess` command is used to create the groups that are used to make up the masking view. Details on creating groups and masking groups are described below.

Creating storage groups

The `symaccess` and the `symsg` commands support storage group operations. When creating a storage group for provisioning, use the `symaccess` command.

You can create a storage group using a range of devices, a list of devices, a device group, a storage group, or a device file. The `symaccess` syntax for creating a storage group is:

```
symaccess -sid SymmID create -name GroupName -type storage
   [devs SymDevStart:SymDevEnd |
         SymDevName[,SymDevName[,SymDevName...]] |
     <-g DgName [-std] [-bcv] [-vdev] [-tgt]> |
     <sg SgName[,SgName1,SgName2,.,SgNamen]>
     <-file DeviceFileName [src] [tgt]>
   [-reserve_id ResvID[,ResvID[,ResvID...]]]]
```

You can add devices at the storage group creation time, but those devices cannot have LUNs assigned to them. The LUNs are assigned by the array when the masking view is created.

You can set Host I/O Limits for a storage group. Host I/O Limits are settings that limit the amount of front-end (FE) bandwidth (MBs) and I/Os per second (IOPs) that can be consumed by a set of array devices over a set of director ports. "Setting Host I/O Limits for storage groups" on page 104 provides all the feature details and restrictions.

In the following example, a range of devices containing storage for a critical application are put into a newly created storage group named `SG_1`:

```
symaccess create -sid 458 -name SG_1 -type storage devs 050:055
```

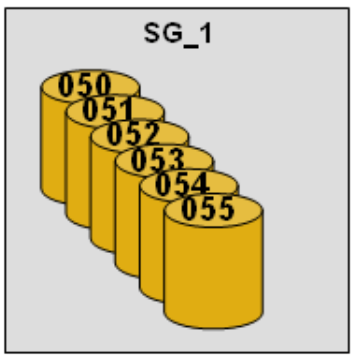


Figure 9 Storage group SG_1

Group names can be up to 64 characters, and are not case sensitive. Names must begin with an alphanumeric character and may contain embedded hyphens and underscore

characters. Group names must be unique per group *type*, but different group types can share the same name. For example, a storage group, a port group, and an initiator group can all have the name `Financial_DB`. However, two storage groups cannot be named `Financial_DB`.

Device reservations are enforced whenever devices are added to a storage group.

The following sections provide more information about storage groups:

"Managing storage groups" on page 321 explains how to add and remove devices, rename a storage group, and delete a storage group.

"Displaying auto-provisioning group information" on page 332 explains the masking view output.

Creating port groups

A port can belong to more than one port group. However, for arrays running HYPERMAX OS 5977 or higher, you cannot mix different types of ports (physical FC ports, virtual FC ports and iSCSI virtual ports) within a single port group. Use the following form to create a port group:

```
symaccess -sid SymmID -name GroupName -type port
  create [-dirport Dir:Port[,Dir:Port...]]
```

In the following example, a new port group, PG_1 is created containing three front-end ports:

```
symaccess create -sid 458 -name PG_1 -type port -dirport 7E:0,7G:1,8F:0
```

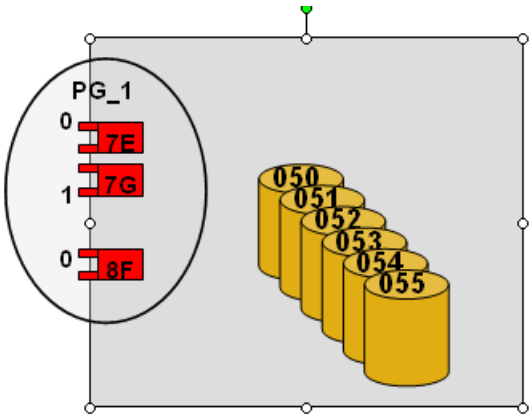


Figure 10 Port group PG_1

The following sections provide more information about port groups:

- "Managing port groups" on page 323 explains how to add and remove ports, rename a port group, and delete a port group.
- "Displaying auto-provisioning group information" on page 332 explains the masking view output.

Creating initiator groups

An initiator group is a container of one or more host initiators (Fibre or iSCSI). Each initiator group can contain up to 64 initiator addresses or 64 child IG names. Initiator groups cannot contain a mixture of host initiators and child IG names.

For arrays running HYPERMAX OS 5977 or higher, you cannot mix different types of initiators (external Fibre Channel WWNs, internal guest Fibre Channel WWNs and iSCSI IQNs) within a single Initiator Group. In addition, all child IG names added to a parent initiator group must contain the same Initiator type.

You can create an initiator group using the HBA's WWN, iSCSI, a file containing WWNs or iSCSI names, or another initiator group name. The `symaccess` syntax for creating an initiator group follows:

```
symaccess -sid SymmID

create -name GroupName -type initiator
      [-consistent_lun]
      [-wwn wwn | -iscsi iscsi |
       -file InitiatorFilename | -ig InitiatorGroupName ]
```

Use the `-consistent_lun` option if the devices of a storage group (in a view) need to be seen on the same LUN on all ports of the port group. If the `-consistent_lun` option is set on the initiator group, Solutions Enabler will make sure that the host LUN number assigned to devices is the same for the ports on the HBA. If this is not set, then the system will choose the first available LUN on each individual port.

In the following example, an initiator group, `IG_1` is created containing one WWN:

```
symaccess create -sid 458 -name IG_1 -type initiator -file IG_1
```

Where the file `IG_1` contains:

```
wwn:210000e08b04daac
```

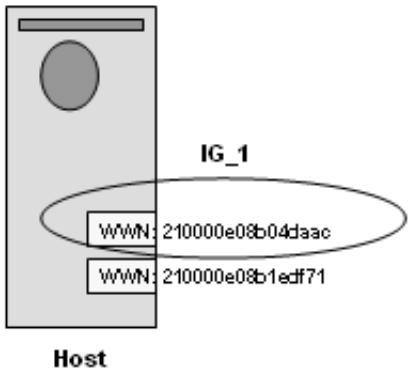


Figure 11 Initiator group IG_1

The following sections provide more information about initiator groups:

- "Managing initiator groups" on page 326 explains how to add and remove initiators, rename an initiator group, and delete an initiator group.
- "Displaying auto-provisioning group information" on page 332 explains the masking view output.

Creating a masking view

A masking view is a container of a storage group, a port group, and an initiator group. When you create a masking view, the devices in the storage group become visible to the host. The devices are masked and mapped automatically.

Volume dynamic addressing is enabled by default. The array assigns the next available LUN address on the FA port when the masking view is created. The LUN assigned on the FA port will not necessarily match the masking LUN.

The `symaccess` syntax for creating a masking view is as follows:

```
create view -sid SymmID -name ViewName -sg StorageGroupName
       -pg PortGroupName -ig InitiatorGroupName
       [ < [-reserve_id ResvID[,ResvID[,ResvID...]]]
       [-lun Addr,Addr,Addr...] [-ckd] [-celerra] [-rp]
```

The groups being used must already exist and contain some entries (the initiator group can be empty) so that a complete view can actually be created.

Note: For arrays running HYPERMAX OS 5977, encapsulated devices that are being used as TimeFinder SnapVX source or target devices cannot be masked to a host.

Use the `-ckd` flag to mask CKD devices. By default, CKD devices are blocked from masking but will be allowed if the devices are already mapped. Celerra devices can be masked without the flag if they are not mapped. Use the `-celerra` flag to unmap Celerra devices. Use the `-rp` flag to map or mask devices that have been tagged for RecoverPoint.

When you create a masking view, if Host I/O Limits have been set for the storage group, they become active with the `symaccess create view` command.

In the following example, a masking view, `MV_1`, is created containing the previously-created storage group, port group, and initiator group:

```
symaccess -sid 458 create view -name MV_1 -sg SG_1 -pg PG_1 -ig IG_1
```

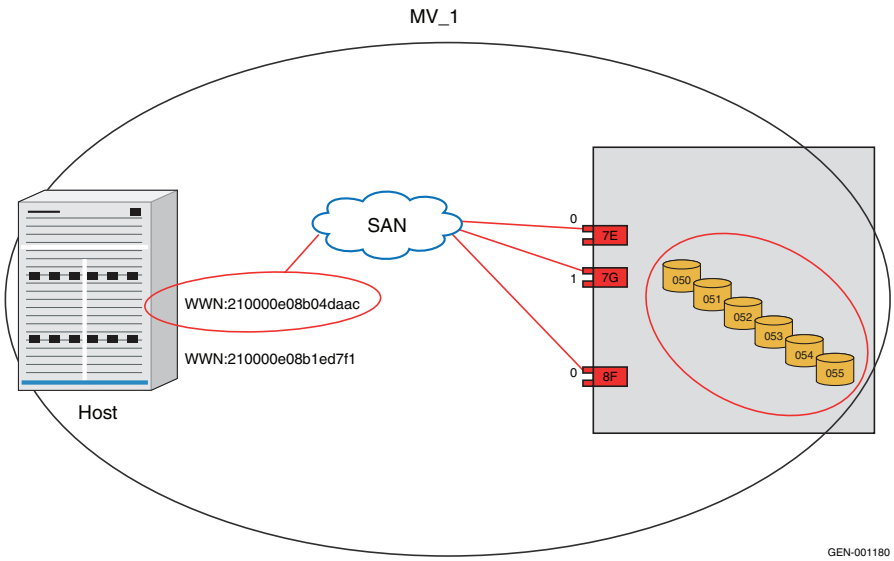


Figure 12 Masking view MV_1

After you have created the masking view, if you need to add additional storage for your application, you can add devices to the storage group and they will be automatically masked and mapped in the masking view.

The same principle applies for adding additional front-end ports or host initiators.

If you have storage that you want to manage from the same host, you can create a second storage group and then create a masking view with the same port group and initiator group. The same number of LUNs can be supplied.

Supplying this value is optional and the corresponding input flag should be supplied when it is given.

The masking view shown in Figure 12 on page 318 is a simple example. In a clustered environment, some devices may be seen by the entire cluster, but gatekeeper devices may only need to be seen by individual hosts. To show the flexibility of the storage provisioning functionality, the next section adds additional groups and another masking view to the example.

Managing masking views

This section explains how to perform the following management operations on maksing views:

Deleting masking views

When a masking view is deleted, all groups in the masking view remain intact. Any device reservations will continue to be enforced when a masking view is deleted.

To delete a masking view, use the following form:

```
symaccess -sid SymmID delete view -name ViewName
```

For example, to delete masking view `MV_1`, on array `458`, enter:

```
symaccess -sid 458 delete view -name mv_1
```

Naming groups and views

Names for storage groups, port groups, initiator groups, and masking views can be up to 64 characters in length. Names must begin with an alphanumeric character and may contain embedded hyphens and underscore characters. Names are not case sensitive. They must be unique per group *type*. In other words, you can have the same name for a storage group, port group, and initiator group, but you cannot use the same name for two storage groups within the same array.

Renaming

Use the `rename -new_name` command to rename a storage group, port group, or an initiator group, using the following form:

```
symaccess -sid SymmID rename -name GroupName -type <storage | port | initiator>
       -new_name NewGroupName
```

Use the `rename view -new_name` command to rename a masking view, using the following form:

```
symaccess -sid SymmID rename view -name ViewName -new_name NewViewName
```

If the new name already exists, an error returns.

Cascaded initiator groups

The renaming of an initiator group will be propagated to the higher group if the group is cascaded.

Backing up and restoring views

The masking views, including storage groups, port groups, and initiator groups can be backed up to a file, and restored from the backup file.

The `backup` and `restore` functionality for `symaccess` provides compact data records.

Use the following syntax to backup the masking views for an array to a file:

```
symaccess -sid SymmID -f BackupFilename [-noprompt]

   backup
   restore [-remove_ckd][-unused_sgs][-disassociate]
```

The `symaccess` command will validate the consistency of the Auto-provisioning data before the `backup` or `restore` actions are performed.

Use the `-noprompt` option to eliminate the prompt for confirmation of the operation. Use the `-remove_ckd` option to skip all CKD devices within the backup, allowing the backup to be restored if the CKD devices are no longer mapped.

The `-disassociate` option disassociates the storage group from a FAST policy if the storage group contains invalid devices for FAST.

The following example shows the message displayed when consistency errors are found during a backup operation:

```
symaccess backup -sid 266 -f aclx_backup -nop

Starting a backup operation.................

There are inconsistencies in the masking database. The operation
   cannot be performed.
```

The following is an example of the displayed message when no consistency errors are found during the backup operation.

```
symaccess backup -sid 266 -f aclx_backup -nop

Starting a backup operation.................

The masking data on Symmetrix 000192600266 was backed up to file
   aclx_backup.
```

Copying groups and views

The `symaccess copy` command provides the ability to copy a storage, port, or initiator group, or a complete masking view from one array to another array. When copying, any child view or cascaded initiator group will be included in the copy action.

Both arrays must be running Enginuity 5874 or higher.

To perform the copy action, use the following form:

```
symaccess -sid SymmID -target_sid SymmID

   copy -name GroupName -type storage
```

```
        [-reserve_id ResvID[,ResvID[,ResvID...]]]

   copy -name GroupName -type initiator | port

   copy -name ViewName view [-ckd] [-celerra] [-rp]
        [-reserve_id ResvID[,ResvID[,ResvID...]]]
```

For example, to copy masking view `mv_1` from array `207` to array `123`, enter:

```
symaccess -sid 207 -target_sid 123 copy -name mv_1 view
```

Use the `-reserve_id` option to include any device reservations, use the `-ckd` option to specify that the view contains CKD devices, use the `-celerra` option to specify that the view contains Celerra devices (the devices will also be mapped), and use the `-rp` option to include devices that have been tagged for RecoverPoint.

Managing storage groups

After creating a storage group, as explained in "Creating storage groups" on page 315, devices can be added to and removed from the storage group. Other actions on a storage group include renaming the group and deleting the group. This section explains how to perform all these actions using the `symaccess` command.

Adding devices

A storage group can contain up to 4k array device numbers, and devices can belong to more than one storage group. When adding devices, specify the device names, a range of devices, a list of devices in a device group, or devices in a device file.

To add devices to an existing storage group, use the following form:

```
symaccess -sid SymmID -name GroupName -type storage
         [-reserve_id ResvID[,ResvID[,ResvID...]]]
         [-ckd] [-celerra] [-rp]

add devs SymDevStart:SymDevEnd [-lun Addr] |SymDevName [-lun Addr] |
         SymDevName,SymDevName,SymDevName...
         [-lun Addr | -lun Addr,Addr,Addr...]

add -g DgName [-std] [-bcv] [-vdev] [-tgt] [-lun Addr]
add -file DeviceFileName [src] [tgt] [-lun Addr]
add sg SgName [,SgName1,SgName2,.,SgNamen]
       [-lun Addr, Addr, Addr...]
```

Device reservations will continue to be enforced when they are added to a storage group. Use the `-ckd` option for adding CKD devices to a storage group. Use the `-celerra` option for adding (and mapping) Celerra devices to a storage group. Use the `-rp` option for adding devices tagged for RecoverPoint to a storage group.

Note: Solutions Enabler supports adding storage groups (`symaccess ad sg`) that have Host I/O Limits set to a parent storage group that is in a view using a port group with FCoE ports.

Note: Enginuity 5876 blocks adding AS/400 and RecoverPoint-tagged devices to a storage group that is in a masking view containing FCoE directors. In addition, a `restore` of AS/400 and RecoverPoint-tagged devices to a storage group that is in a masking view containing FCoE directors is not supported. When detected, the AS/400 and RecoverPoint-tagged devices will be removed prior to completing the `restore` action.

Designating the LUN address

When devices are added at the storage group creation time, do not specify a LUN address. The LUN address is determined when the masking view is created.

LUN addresses should only be supplied if the storage group is already contained within a view. In this case, a single LUN can be given, or one for each device range. If the LUN address is not specified, the array will assign the LUN address.

Removing devices

A storage group should not be deleted until all the devices have been removed. To remove devices from a storage group, or child storage groups, use the following form:

```
symaccess -sid SymmID -name GroupName -type storage
  [-reserve_id ResvID[,ResvID[,ResvID...]]] [-force]
   [-unmap [-celerra]] [-ckd] [-rp]

remove devs SymDevStart:SymDevEnd | SymDevName |
    SymDevName,SymDevName,SymDevName...

remove -g DgName [-std] [-bcv] [-vdev] [-tgt]

remove -file DeviceFileName [src] [tgt]

remove sg SgName[,SgName1,SgName2,.,SgNamen]
```

For example, to remove the BCV devices in device group `Prog2` from storage group `SG_Prod` on array `458`, enter:

```
symaccess -sid 458 -name SG_Prod -type storage remove -g Prog2 -bcv
```

Note: A storage group cannot be completely emptied if it is associated with a masking view.

Renaming a storage group

Use the `rename -new_name` command to rename a storage group, port group, or an initiator group, using the following form:

```
symaccess -sid SymmID rename -name GroupName -type <storage | port |
   initiator> -new_name NewGroupName
```

Deleting storage groups

A storage group should be empty before being deleted, unless using the `-force` option.

```
symaccess -sid SymmID

delete <view -name ViewName [-unmap [-emulation celerra]]
       [-reserve_id ResvID[,ResvID[,ResvID...]]] > |
```

```
< -name GroupName -type <storage
   [-reserve_id ResvID[,ResvID[,ResvID...]]] |
port | initiator> [-force] > [-noprompt][-emulation ckd]
```

For example, to delete storage group SG_1 from array 458, enter:

```
symaccess -sid 458 -name sg_1 -type storage delete
```

Note: A storage group cannot be deleted if it is associated with a masking view or is in use by a FAST policy. To delete both a storage group and a masking view, delete the masking view first, then delete the storage group.

Use the -unmap option to unmap the devices from the ports contained in the port group of the view. If a device is mapped to the same port through another view, it will not be unmapped until the second masking view is deleted.

Listing and showing storage group information

To view storage group information, use the following form:

```
symaccess -sid <SymmID> [-offline] | -file <backup_filename>

     list -type storage
        [-devs <SymDevStart:SymDevEnd |
        SymDevName | <SymDevName,SymDevName...>>]
        [-name GroupName] [-v | -detail]

     show GroupName -type storage
```

You can request the information from the array or from the backup file. In addition, you can request a specific device range or list of devices.

Managing port groups

Port groups contain director and port identification and belong to a masking view. Ports can be added and removed. When a port group is no longer associated with a masking view, it can be deleted. In addition, CHAP authentication can be enabled and disabled on port groups. This section details these port management operations.

The symaccess control operations add, remove, backup, and restore are modified to support the extended FA director port range.

Adding ports

Ports and iSCSI targets can be added to an existing port group by specifying the name and type of the group and the director port or iSCSI target information, using the following form:

```
symaccess -sid SymmID -name GroupName -type port
       [-ckd][-celerra] [-rp]
   add -dirport Dir:Port[,Dir:Port[,Dir:Port...]]
   add -iscsi_tgt iSCSI [,iSCSI...}
```

Use the -ckd, -celerra, or -rp options to specify the device types.

To add port 1 of Fibre director 16D to the existing port group PG_1 on array 245, enter:

```
symaccess -sid 245 -name PG_1 -type port add -dirport 16D:1
```

Removing ports

To remove a port or an iSCSI target, use the following form:

```
symaccess -sid SymmID -name GroupName -type port
   [-ckd][-unmap][-force][-celerra] [-rp]]

remove -dirport Dir:Port[,Dir:Port[,Dir:Port...]]
remove -iscsi_tgt iSCSI [,iSCSI...]
```

For example, to remove port 1 of Fibre director 16D from the existing port group PG_1 on array 245, enter:

```
symaccess -sid 245 -name PG_1 -type port remove -dirport 16D:1
```

Note: A port group cannot be completely emptied if it is associated with a masking view.

Use the `-unmap` option to unmap all devices in the storage group associated with the view. Use the `-force` option to remove the port.

On arrays running HYPERMAX OS 5977, only a single front-end emulation of each type (FA, EF, etc.) can be assigned to each director, however each of these emulations can be assigned a variable number of physical ports (up to 32, numbered from 0 - 31). The `symaccess` control operations `copy -type port` or `copy view` copies the extended FA director port range (0 to 31) target array if the target array is running HYPERMAX OS 5977 or higher.

Deleting port groups

To delete a port group, use the following form:

```
delete <view -name ViewName> [-force] |
   -name GroupName -type <storage | port | initiator > [-noprompt]
```

Note: A port group cannot be deleted if it is associated with a masking view.

For example, to delete port group PG_1 on array 245, enter:

```
symaccess -sid -name PG_1 -type port delete
```

Copying a port group from array to array

Use the following form to copy a port group from one array to another:

```
symaccess -sid SymmID -target_sid SymmID
     copy -name GroupName -type port
```

Listing and showing port group information

Use the following form to display port group information:

```
symaccess -sid <SymmID> [-offline] | -file <backup_filename>
     list -type port [-dirport Dir:Port]
           [-name GroupName] [-detail | -v]
     show GroupName -type port
```

Locking down a Fibre Channel ID

Fibre Channel ID (FCID) lockdown is a security feature that limits host device access by adding Fibre Channel ID information of a switch within a fabric to device access records in the login history table. This feature handles WWN spoofing and the threat it poses to your networked systems in a shared (same director port) storage port configuration.

For example, to implement the Fibre Channel ID lockdown feature on Fibre Channel 021300 for director 16A, port 0, enter:

```
symaccess -sid 018 set lockdown on 021300
```

This feature lets you set the Fibre Channel ID (FCID) of the WWN of the HBA you want to protect. The FCID is then added to the database record for the WWN of the specified HBA with the specified director and is locked. Once a Fibre Channel ID is locked, no user with a spoofed WWN can log in. If a user with a spoofed WWN is already logged in, that user loses all access through that HBA.

⚠CAUTION

When an HBA logs in to a director port, the Fibre Channel ID accompanies it, telling the director port where to send its response. By specifying Fibre Channel ID information of the switch, the valid physical path through the SAN for a particular HBA is locked down. Only an HBA with a Fibre Channel ID that matches the FCID specified in the device masking record is able to log in to the storage port. It is recommended that at least two HBAs be available on the administrator host. If one HBA becomes locked out, the host will have access through the other HBA and can correct the record in the database.

Lockdown steps

To find the Fibre Channel ID, lock it down, verify that it is locked down, and then force the change to take effect, use the following procedure:

1. Find the WWN. If the device is visible, run `symaccess list hba` to find the device path of the HBA you want to protect.
2. Find the Fibre Channel ID value by referring to "Finding the FCID of a switch" on page 378.
3. Run `symaccess set lockdown` to on with the FCID of the Fibre Channel ID you found in step 2.
4. To finalize the change, either reboot the host or pull the cable from the director and then replace the cable.

Managing initiator groups

This section describes how to manage initiator groups.

Adding initiators

Initiators can be added to an existing initiator group by specifying the initiator type (`-wwn` or `-iscsi`), the initiator group name, or by using an input file. The following is the syntax for adding one or more initiators to an initiator group:

```
symaccess -sid SymmID -name GroupName -type initiator
        -wwn wwn | -iscsi iscsi | -ig InitiatorGroupName |
         -f InitiatorFilename
   add
```

Individual initiators

For example, to add initiator `-wwn 10000000c94ef69c` to the initiator group `IG_1` on array `245`, enter:

```
symaccess -sid 245 -name IG_1 add -type initiator -wwn 10000000c94ef69c
```

Using an output file

When using an input file, each initiator must be placed on a new line and start with either `WWN:` or `iSCSI:` or `IG:`, depending on the type of the initiator or initiator group name. The following is an example of the format for an initiator file:

```
WWN:10000000c94ef69c
iSCSI:iscsiname
IG:IgName
#WWN:10000000c94ef69d
```

If the format of the initiator does not match the label at the start of the line, the file returns an error. A commented line, which the system ignores, is specified by placing the pound sign (`#`) at the beginning of a line.

Cascaded initiator groups

An initiator group can be added to another initiator group, only if it does not contain any initiator groups.

The following scenario describes cascaded initiator groups:

- HOST1 contains WWN1 & WWN2, which are added to IG_1.
- HOST2 contains WWN3 & WWN4, which are added to IG_2.
- IG_3 is created and contains IG_1 & IG_2.

In this example, gatekeeper devices for HOST1 can be assigned to IG_1, while different gatekeeper devices for HOST2 can be assigned to IG_2. The application devices needed by both hosts can be assigned to IG_3.

Note: If using the Volume Set Addressing flag, both the parent and child initiator group must have the flag.

Removing initiators

The `symaccess remove` command removes specified initiators from an initiator group.

An initiator group that is currently associated with a masking view will be allowed to be emptied and the view will remain along with the other two groups. The following is the syntax for removing an initiator from an initiator group:

```
symaccess -sid SymmID -name GroupName -type initiator
        -wwn wwn | -iscsi iscsi | -ig InitiatorGroupName
        | -f InitiatorFilename [-login]
   remove
```

For example, to remove initiator `-wwn 10000000c94ef69c` from the initiator group `IG_1` on array `245`, enter:

```
symaccess -sid 245 -name IG_1 remove -type initiator -wwn 10000000c94ef69c
```

Add the `-login` option to the command to remove the initiator from the array's login history table.

Deleting initiator groups

To delete an initiator group, use the following form:

```
delete <view -name ViewName> [-force] |
   -name GroupName -type <storage | port | initiator > [-noprompt]
```

For example, to delete initiator group `IG_1` on array 245, enter:

```
symaccess -sid -name IG_1 -type initiator delete
```

Note: An initiator group cannot be deleted if it is associated with a masking view.

Setting initiator group flags

This feature allows you to set all the available port flags at the initiator group level. The following flags are supported for initiator groups:

```
Volume_Set_Addressing    [V]
Common_Serial_Number     [C]
Disable_Q_Reset_on_UA    [D]
Environ_Set              [E]
Avoid_Reset_Broadcast    [ARB]
AS400                    [AS4]
OpenVMS                  [OVMS]
SCSI_3                   [SC3]
SPC2_Protocol_Version    [SPC2]
SCSI_Support1            [OS2007]
```

Note: Modifications to the AS400 and Common_Serial_Number initiator group port flags are not allowed on arrays running HYPERMAX OS 5977 or higher.

A flag cannot be set for the group if it conflicts with any initiator in the group. After a flag is set for a group, it cannot be changed on an initiator basis.

To set an override flag for an initiator group, use the following form:

```
symaccess -sid SymmID -name GroupName -type initiator
     set ig_flags <on <Flag> <-enable |-disable> | off [Flag]>
```

Where:

`on` — Turns on the specified initiator group port *`Flag`* override and allows you to set the status of that flag to either `enabled` or `disabled`.

`off` — Turns off the initiator group port *`Flag`* override.

`enable` — Sets the status of the initiator group port *`Flag`* to `enabled`. The status can only be set when the initiator group port flag override setting value is `on`.

`disable` — Sets the status of the initiator group port *`Flag`* to `disabled`. The status can only be set when the initiator group port flag override setting value is `on`.

For example, to set the `OS2009 [OS2009]` for the initiator group `my_ig` on array `266`, enter:

```
symaccess -sid 266 -type init -name my_ig set ig_flags on OS2009 -enable
```

Use the `show` command with the `-detail` option to view the flag from the previous command:

```
symaccess -sid 266 show my_ig -type init -detail
```

Output similar to the following example displays:

```
Symmetrix ID          : 000192600266

Initiator Group Name  : my_ig
Last updated at       : 10:52:15 AM on Wed Mar 31,2010

Port Flag Overrides   : Yes
  Enabled             : OS2009(OS2009)
                      Common_Serial_Number(C)
  Disabled            : Avoid_Reset_Broadcast(ARB)
Consistent Lun        : No

Originator Port wwn   : 1234567822446688
User-generated Name   : 1234567822446688/1234567822446688
FCID Lockdown         : No
Heterogeneous Host    : No
Port Flag Overrides   : Yes
Enabled               : OS2009(OS2009)
                      Common_Serial_Number(C)
Disabled              : Avoid_Reset_Broadcast(ARB)
CHAP Enabled          : N/A
Type                  : Fibre
```

Note: The Write Protect Bypass flag is only supported on arrays running Enginuity 5875. The flag is not supported in Enginuity 5876. If a backup file contains an initiator group with the Write Protect Bypass flag set, a `restore` operation on an array running Enginuity 5876 will clear the flag from the initiator group and all its initiators.

Setting HBA flags

For arrays running Enginuity 5874 and higher, the HBA port flags are set on a per initiator basis. The HBA must belong to an initiator group. This feature allows specific host flags to be enabled and disabled on the director port.

Note: Setting HBA port flags replaces setting the heterogeneous host configuration flags. To switch to setting HBA port flags, the heterogeneous host configuration must be disabled for a given HBA and all flags must be reset.

To set (or reset) the HBA flags, use the following form:

```
symaccess -sid SymmID -wwn wwn | -iscsi iscsi

set hba_flags <on <flag,flag,flag...> <-enable |-disable> |
       off [flag,flag,flag...]>
```

Where:

`hba_flags` — Sets the record in the database to hold information on the HBA port setting that may differ than the current setting on the FA.

`on | off` — Turns HBA flags on or off.

`flag` — Specifies the overridden HBA port flags from the values [in brackets]:

Supported HBA port flags:

```
AS400 [AS4]
Avoid_Reset_Broadcast [ARB]
Common_Serial_Number  [C]
Disable_Q_Reset_on_UA ] [D]
Environ_Set [E]
OpenVMS [OVMS]
SCSI_3  [SC3]
SCSI_Support1 [OS2007]
SPC2_Protocol_Version [SPC2]
```

Supported initiator group port flags:

```
AS400 [AS4]
Avoid_Reset_Broadcast [ARB]
Common_Serial_Number  [C]
Disable_Q_Reset_on_UA ] [D]
Environ_Set [E]
OpenVMS [OVMS]
SCSI_3  [SC3]
SCSI_Support1 [OS2007]
SPC2_Protocol_Version [SPC2]
Volume_Set_Addressing [V]
```

Note: Modifications to the AS400 and Common_Serial_Number initiator group port flags are not allowed on arrays running HYPERMAX OS 5977 or higher.

`-enable` — Enables the specified HBA port flag(s) on a per initiator basis.

`-disable` — Disables the specified HBA port flag(s) on a per initiator basis.

For example, to turn on HBA flags and enable the `Common_Serial_Number` and `SCSI_3` flags, and disable the `Disable_Q_Reset_on_UA` flag, on an HBA with the WWN ID `210000e08b0995b7` for array 031, director 16A port 0, enter:

```
symaccess -sid 031 set hba_flags on C,SC3 -enable -wwn 210000e08b0995b7-dir 16A -p 0
symaccess -sid 031 set hba_flags on D -disable -wwn 210000e08b0995b7-dir 16A -p 0
```

The `symaccess show -detail` output has been modified to display the flags that are turned on and off for each HBA initiator that has this feature enabled, as shown in the following example:

```
symaccess -sid 237 -type initiator -detail show Prod1

Symmetrix ID            : 000190300237
Last updated at         : 08:46:54 AM on Tue Jul 29,2008

Initiator Group Name    : Prod1
```

```
Originator Port wwn  : 10000000c94ef69c
   User-generated Name  : api196/10000000c94ef69c
   FCID Lockdown        : No
   Heterogeneous Host   : No
   Port Flag Overrides  : No
   Type                 : Fibre

   Originator Port wwn  : 5006016839a00c5c
   User-generated Name  : 5006016839a00c5c/5006016839a00c5c
   FCID Lockdown        : No
   Heterogeneous Host   : No
   Port Flag Overrides  : No
   Type                 : Fibre

   iSCSI Name           : Symm_iScsi
   User-generated Name  : iScsi_node_alias/iScsi_port_alias
   FCID Lockdown        : N/A
   Heterogeneous Host   : No
   Port Flag Overrides  : No
   Type                 : iScsi

Group Name           : IniGrp
   User-generated Name  : N/A
   FCID Lockdown        : N/A
   Heterogeneous Host   : N/A
   Port Flag Overrides  : N/A
   Type                 : Initiator Group
```

Replacing an HBA

If a host adapter fails, or needs replacement for any reason, you can replace the adapter and assign its set of devices to a new adapter by using the `replace` action in the following form:

```
symaccess replace -wwn wwn -new_wwn NewWWN [-noprompt]
```

```
symaccess replace -iscsi iscsi -new_iscsi NewiSCSI [-noprompt]
```

To swap HBAs:

1. Run `symaccess list logins` to view the old WWN/iSCSI HBAs.
2. Swap the HBA boards.
3. Run `symaccess list hba` or `discover` to view the new initiator (for example WWN).
4. Run `symaccess replace` to substitute a new WWN for all occurrences of the old WWN. For example, to replace old WWN `20000000c920b484` with new WWN `20000000c920b393`:

```
symaccess -sid 814 replace -wwn 20000000c920b484 -new_wwn 20000000c920b393
```

5. Run `symaccess discover -rename` to establish the new AWWN and assign an AWWN to the new HBA in the login history table.

Renaming an HBA

To rename the alias for a specified initiator within a group, use the following form:

```
symaccess -sid SymmID
   rename -wwn wwn -alias alias | -iscsi iscsi -alias alias
```

Using CHAP authentication

CHAP (Challenge Handshake Authentication Protocol) allows you to manage a credential name and a CHAP secret, which are similar to a username and a password, though more secure than the standard Password Authentication Procedure (PAP).

To enable CHAP on an iSCSI initiator, use the following form:

```
symaccess -sid SymmID -iscsi iscsi enable chap
```

To enable CHAP on a director and port, use the following form:

```
symaccess -sid SymmID [-dirport Dir:Port]
enable chap
```

To set the CHAP credential and secret, use the following form:

```
symaccess -sid SymmID -dirport Dir:Port | -iscsi iscsi
   set chap -cred Credential -secret Secret
```

To enable CHAP on a specific director and port, use the following form:

```
symaccess -sid SymmID [-dirport Dir:Port] enable chap
```

To disable CHAP on a specific director and port, use the following form:

```
symaccess -sid SymmID [-dirport Dir:Port] disable chap
```

To delete CHAP from a specific director and port, use the following form:

```
symaccess -sid SymmID [-dirport Dir:Port] delete chap
```

Displaying CHAP information

The following is an example for displaying the CHAP information.

```
symaccess list chap -sid 001

Symmetrix ID            : 000197100001

Director Identification : SE-9G
Director Port           : 310
iSCSI Target Name       : iqn.1992-04.com.emc:sn.11121318

Protocol                : CHAP

Identifier                Type   State     Credential
------------------------  -----  --------  ------------------------
SE-9G:310                 N/A    ENABLED   credential

symaccess list chap -sid 001 -v

Symmetrix ID            : 0001971000001

Director Identification : SE-9G
Director Port           : 310
iSCSI Target Name       : iqn.1992-04.com.emc:sn.11121318

Protocol                : CHAP
   Identifier           : SE-9G:310
   Identifier Type      : N/A
   State                : ENABLED
   Last updated at      : 09:58:27 AM on Fri Apr 17,2015
   CHAP Credential      : credential
```

Verifying the database

Use the `symaccess verify` command to verify that the Auto-provisioning database is consistent. Any inconsistencies display in the command output. This command can also be used with a backup file. Add the `-log` option for reporting the inconsistencies in a log file and on the screen.

To verify the ACLX database, use the following form:

```
symaccess -sid SymmID | -file BackupFileName
  verify [-log]
```

The following example shows the message displayed when no consistencies are found:

```
symaccess -sid 266 verify

Starting a verify operation.................
The auto provisioning database is consistent
```

The next example shows verification of a database backup file with inconsistencies:

```
symaccess -file /tmp/bkup1.file verify

Starting a verify operation................

Found SG 'stor_grp1' to contain the view flag but didn't find a
   matching view

Found IG 'init_grp1' contains invalid initiator records

Found masking view 'mask_view1' with parent IG 'init_grp1' but no
   masking records for the child IG 'child_grp1' are present

There are inconsistencies in the auto provisioning database
```

Displaying auto-provisioning group information

With Solutions Enabler, when you list information without using the `-detail` option, any column without data will not display. If the `-detail` option is provided, the column without data will display a dash (`-`).

This section shows the output for the masking views using the following form:

```
symaccess -sid SymmID [-offline] | -file BackupFilename

   list [-name GroupName] [-v]

   list -type <storage [-devs <SymDevName[:SymDevName]>] | port
         [-dirport Dir:Port] | initiator [-wwn wwn |
         -iscsi iscsi]> [-name GroupName] [-detail | -v]

   list devinfo [-ig InitiatorGroupName]

   list view [-name ViewName][-v][-detail]

   show GroupName -type <initiator [-detail] | port | storage>

   show view ViewName [-ig ChildInitiatorGroupName]

symaccess -sid SymmID | -file BackupFilename

   list chap [-dirport Dir:Port][-v]
```

```
symaccess -sid SymmID

   list assignment [-v] -devs <SymDevStart:SymDevEnd |
        SymDevName | <SymDevName,SymDevName...>

   list no_assignments [-dirport Dir:Port]
```

A new `-offline` option is available for the `symaccess list` and `show` commands. When the `-offline` option is used, the command reports the data from the symapi configuration database file and not from the array. Alternatively, the SYMCLI_OFFLINE environment variable can be set to 1 to enable the offline mode for reporting.

```
symaccess -sid SymmID [-offline] | -file BackupFilename

   list [-name GroupName] [-v]
   ...
```

For arrays running HYPERMAX OS 5977, only a single front-end emulation of each type (FA, EF, etc.) can be assigned to each director, however each of these emulations can be assigned a variable number of physical ports (up to 32, numbered from 0 - 31). The following `symaccess` reporting commands are modified to support the extended FA director port range:

- For `symaccess show -type port`, `show view`, and `symaccess show backupfile view` output, the Director Identification section is modified to support reporting for the extended port range. In addition new fields are added: the WWN port name, which displays the Fibre channel director port WWN, and the iSCSI name, which displays the name of the iSCSI director port.
- For `symaccess list hba, symaccess list devinfo output,` and `symaccess list assignments` the Dir:Port field is modified to support reporting for the extended port range.
- The `symaccess list chap` output is modified to support reporting for the extended port range. In addition, a new field is added, the iSCSI Target Name, which displays the iSCSI name for the iSCSI director port.
- For `symaccess list no_assigments and symaccess list logins` output, the Director Port field is modified to support reporting for the extended port range.

Note: For arrays running HYPERMAX OS 5977, the `symaccess list no_assignments` output will report that “All devices available on this director/port are assigned” for ACLX enabled ports.

Showing the last update time

The displays for `symaccess` return the current array time and the time when the group was modified in seconds. This delta time reports the last updated time of a provisioning group, or any related groups of the selected group.

Displaying masking views

The following example shows the command and output for listing all masking views on array 237:

```
symaccess -sid 237 list view

Symmetrix ID            : 000190300237

Masking View Name   Initiator Group Port Group Storage Group
------------------- ---------------- --------- --------------
View1                IG_1             PG_1     SG_1
View2                WinHost          PG_5     Acct
```

...

Use the `-detail` option with the `list` command to display all the details of a view, including the child initiator groups.

The `symaccess show view` command displays the last update time of the masking view. The groups that make up a masking view include the port group, the initiator group and any child initiator groups, the storage group and any child storage groups. For example:

```
symaccess -sid 0086 show view app_view1 -detail

Symmetrix ID                : 000197100001

Masking View Name           : app_view1
Last update time            : 03:59:50 PM on Tue Aug 26,2014
View last update time       : 09:34:25 AM on Thu Sep 04,2014

Initiator Group Name        : app_parent_ig1

   Host Initiators
     {
       IG   : app_child_ig1
       IG   : app_child_ig2
     }

Port Group Name             : app_2e_10

   Director Identification
     {
        Director
      Ident  Port   WWN Port Name / iSCSI Target Name
      ------ ----
   -------------------------------------------------------
      FA-2E   010 5000097370011234
     }

Storage Group Name          : app_parent_sg1

   Number of Storage Groups : 3
   Storage Group Names      : app_csg1                        (IsChild)
                              app_csg2                        (IsChild)
                              app_csg3                        (IsChild)

Masking View Name           : app_view1
Last update time            : 03:59:50 PM on Tue Aug 26,2014
View last update time       : 09:34:25 AM on Thu Sep 04,2014

Initiator Group Name        : app_child_ig1 *

   Host Initiators
     {
       WWN  : 1212121212121211
              [alias: 1212121212121211/1212121212121211]
```

```
     }

Port Group Name             : app_2e_10

   Director Identification
     {
        Director
      Ident  Port   WWN Port Name / iSCSI Target Name
      ------ ----
   -------------------------------------------------------
      FA-2E   010 5000097370011234
     }

Storage Group Name          : app_parent_sg1

   Number of Storage Groups : 3
   Storage Group Names      : app_csg1                        (IsChild)
                              app_csg2                        (IsChild)
                              app_Csg3                        (IsChild)

Sym                                          Host
Dev     Dir:Port  Physical Device Name     Lun   Attr  Cap(MB)
------  --------  -----------------------  ----  ----  -------
00040   02E:010   Not Visible                 1           4125
00041   02E:010   Not Visible                 2           4125
00042   02E:010   Not Visible                 3           4125
00043   02E:010   Not Visible                 4           4125
00044   02E:010   Not Visible                 5           4125
00045   02E:010   Not Visible                 6           4125
                                                         -------
Total Capacity                                             24750

Masking View Name           : app_view1
Last update time            : 03:59:50 PM on Tue Aug 26,2014
View last update time       : 09:34:25 AM on Thu Sep 04,2014

Initiator Group Name        : app_child_ig2 *

   Host Initiators
     {
       WWN  : 1212121212121212
              [alias: 1212121212121212/1212121212121212]
     }

Port Group Name             : app_2e_10

   Director Identification
     {
        Director
      Ident  Port   WWN Port Name / iSCSI Target Name
      ------ ----
   -------------------------------------------------------
      FA-2E   010 5000097370011234
     }

Storage Group Name          : app_parent_sg1

   Number of Storage Groups : 3
   Storage Group Names      : app_csg1                        (IsChild)
                              app_csg2                        (IsChild)
                              app_csg3                        (IsChild)

Sym                                          Host
Dev     Dir:Port  Physical Device Name     Lun   Attr  Cap(MB)
------  --------  -----------------------  ----  ----  -------
00040   02E:010   Not Visible                 1           4125
```

```
00041   02E:010   Not Visible                2           4125
00042   02E:010   Not Visible                3           4125
00043   02E:010   Not Visible                4           4125
00044   02E:010   Not Visible                5           4125
00045   02E:010   Not Visible                6           4125
                                                       -------
Total Capacity                                          24750


* Denotes a cascaded Initiator Group within the specified Masking View
```

Viewing group details

Storage group, port group, and initiator group details can be displayed using the `symaccess list` and `symaccess show` commands, as follows:

```
symaccess -sid SymmID [-offline] | -f BackupFilename

   list [-name GroupName] [-v]

   list -type <storage [-devs SymDevName <:SymDevName>] | port
         [-dirport Dir:Port] | initiator [-wwn wwn |
         -iscsi iscsi]> [-v] [-name GroupName]

   show GroupName -type <storage | port | initiator [-detail]>
   show view ViewName
```

An optional group name may be supplied as input and only the details on that group return.

An optional director and port number may be supplied and only the port groups that belong to the director and port return.

To display a list of the initiator groups, in verbose mode, use the following form:

```
   list -type initiator [-wwn wwn | -iscsi iscsi] -v
```

To display a specific initiator group, in verbose mode, use the following form:

```
   list -name GroupName -v
```

To display the details of a specific initiator group, use the following form:

```
   show GroupName -type initiator -detail
```

To display information about specific devices, or device ranges, use the following form:

```
   symaccess -sid SymmID [-offline] | -f BackupFilename
   list -type <storage [-devs SymDevName <:SymDevName>]
   [-name SgName] [-detail | -v]
```

The `symaccess list -type storage -v` command output displays the time since the last update of the storage group or any child storage group.

For example:

```
   symaccess list -type storage -v

   Symmetrix ID           : 000195700601

   Storage Group Name     : backup_storage
   Device Count           : 1
   Storage Group Count    : 0
   Masking View Count     : 0
   Last update time       : 12:32:26 AM on Fri Apr 06,2012
```

```
   Group last update time : 12:32:26 AM on Fri Apr 06,2012
   Masking View Names     : None

   Storage Group Name     : Mkt_storage (IsParent)
   Device Count           : 0
   Storage Group Count    : 1
   Masking View Count     : 0
   Last update time       : 08:10:22 PM on Tue Feb 14,2012
   Group last update time : 10:32:26 AM on Fri Apr 06,2012
   Masking View Names     : Mkt_view *
    . . .
   * Denotes Masking Views through a cascaded group
```

The `symaccess list view -detail` command displays the last update time of the masking view or any of the groups that make up the masking view

For example:

```
symaccess list view -detail

Symmetrix ID                : 000195700601

Masking View Name           : HR_view
Last update time            : 04:30:24 AM on Tue Apr 03,2012
View last update time       : 04:35:36 AM on Tue Apr 03,2012

Initiator Group Name        : HR_hosts

   Host Initiators
     {
       IG   : HR_host_GrpA
     }

Port Group Name             : HR_ports

   Director Identification
     {
       FA-15E:1
     }

Storage Group Name          : HR_storage

   Number of Storage Groups : 0
   Storage Group Names      : None

Sym                                     Host
Dev     Dir:P  Physical Device Name     Lun   Attr  Cap(MB)
------  -----  ----------------------  ----  ----  -------
01C27   15E:1  Not Visible                 1         300000
                                                     -------
Total Capacity                                        300000
```

Viewing device assignments

To display the assignments for one or more devices, use the following form:

```
symaccess -sid SymmID list assignments
      -devs SymDevStart:SymDevEnd | SymDevName |
           SymDevName, SymDevName... [-v]
```

For example, to list the assignments for device range `20:22` and device `24` on array `120`, enter:

```
symaccess -sid 120 list assignments -devs 20:22,24

Symmetrix ID          : 000192600120

Device  Identifier        Type   Dir:P
------  ----------------  -----  ----------------
00020   10000000c9594dce  FIBRE  FA-7E:1
        210000e08b04daac  FIBRE  FA-7E:1
        210000e08b1ed7f1  FIBRE  FA-7E:1
00021   10000000c9594dce  FIBRE  FA-7E:1
        210000e08b04daac  FIBRE  FA-7E:1
        210000e08b1ed7f1  FIBRE  FA-7E:1
00022   10000000c9594dce  FIBRE  FA-7E:1
        210000e08b04daac  FIBRE  FA-7E:1
        210000e08b1ed7f1  FIBRE  FA-7E:1
00024   10000000c9594dce  FIBRE  FA-7E:1
        210000e08b04daac  FIBRE  FA-7E:1
        210000e08b1ed7f1  FIBRE  FA-7E:1
```

Listing no assignments

To display a list of devices with no assignments, use the following form:

```
symaccess -sid SymmID list no_assignments [-dirport DirNum:PortNum]
```

For example, to list the devices without assignments for array `120`, enter:

```
symaccess -sid 120 list no_assignments

Output similar to the following example displays:
Symmetrix ID             : 000192600120

Director Identification  : FA-7F
Director Port            : 0
ACLX Enabled             : No

No devices were found for this director/port

Director Identification  : FA-7F
Director Port            : 1
ACLX Enabled             : Yes

Devices not yet assigned :

00030
00031
00032
00033
00034
...
```

Listing initiator group devices

To view all the devices masked to an initiator group, use the following form:

`symaccess -sid SymmID list devinfo [-ig InitiatorGroupName]`

Initiator group displays with Solutions Enabler include the child initiator group device information when adding the `-detail` option.

Viewing the HBA alias name

Solutions Enabler can display the alias name with up to 32 characters each for the alias node name and the alias port name, as shown in the following example command and output:

```
symaccess show host1082_view -sid 001 view

Symmetrix ID                : 000197100001
Masking View Name           : host1082_view
Last updated at             : 12:51:37 PM on Tue May 13,2014
View last update time       : 01:37:41 PM on Thu Apr 02,2015

Initiator Group Name        : hba1_2_3

   Host Initiators
     {
       WWN    : 10000000C99DE136
                [alias: api019010000000C99DE136/api019010000000C99DE136]
       ISCSI  : iqn.2002-06.com.host1082
                [alias: api1082/api1082]
     }

Port Group Name             : host1082_ports

   Director Identification
     {
         Director
       Ident  Port   WWN Port Name / iSCSI Target Name
       ------ ---- -------------------------------------------------------
       FA-3E   001 5000097300092110
       SE-9H   000 iqn.1992-04.com.emc:5000097300092190
       SE-10G  000 iqn.1992-04.com.emc:5000097300092194
     }

Storage Group Name          : application_sg

   Number of Storage Groups : 0
   Storage Group Names      : None

Sym                                       Host
Dev    Dir:Port  Physical Device Name     Lun   Attr  Cap(MB)
-----  --------  -----------------------  ----  ----  -------
00083   3E:001   Not Visible                 1              6
        9H:000   Not Visible                 1              6
       10G:000   Not Visible                 1              6
                                                       -------
Total Capacity                                               6
```

CHAPTER 11
Masking Devices (Enginuity 5773 or lower)

This chapter describes how to confine host access to array devices using the SYMCLI device masking commands for DMX arrays running Enginuity 5773 or lower.

- Overview 342
- Supported topologies 344
- Using the device masking database 345
- Configuring device masking 345
- Identifying configuration components 346
- Initializing and updating the database 347
- Enabling authentication 347
- Recommendations for activating the configuration 363
- Discovering host HBAs 364
- Adding masked devices 365
- Removing masked devices 368
- Maintaining the device masking database 369
- Viewing the VCMDB 372
- Managing HBA initiators 374
- Managing the Fibre Channel-to-host interface 376

Overview

SYMCLI device masking provides the ability to assign and mask access privileges of host bus adapters (HBAs) to DMX directors and devices by associating one or more devices with an HBA-to-FA connection that you define in the DMX-based device masking database, known as the VCMDB. The VCMDB maintains all access records for an array and the DMX array monitors host access to resolve any conflicts that might arise from multiple hosts having visibility to the same devices.

The device masking commands can also be used to configure heterogeneous hosts with shared access to the same FA port, which is useful in an environment with different host types. However, you can also use Fibre Channel ID lockdown security to protect an HBA from predatory WWN spoofing.

VMAX Family arrays running Enginuity 5874 or higher do not use the `symmask` and `symmaskdb` commands described in this chapter. For Enginuity 5874 or higher, the `symaccess` command is used for device masking. For more information on using the `symaccess` command for VMAX Family arrays, refer to Chapter 10, "Masking Devices with Auto-Provisioning Groups."

Host access process

When a host attempts to access a DMX storage device, the host HBA initiator name (supplied when the host logs in to the fabric or arbitrated loop) is passed to the DMX director port.

The DMX records the connection, stores the initiator name in a *login history table* in its memory, and thereafter grants access to only the devices that are available to that initiator through that director port (as specified in the device masking VCMDB). However, if CHAP authentication is enabled in a native iSCSI topology, the VCMDB will first check the credential and secret before granting access.

The host HBA port then sends I/O requests directed at particular DMX devices to the director port. Each request includes the identity of the requesting HBA (from which its WWN or iSCSI can be determined) and the identity of the requested device, with its director and logical unit number (LUN).

The software that runs on a DMX system processes each I/O request to verify that the HBA is allowed to access that device. Any request for a device that an HBA does not have access to, returns an error to the host.

Managing access for unsupported host platforms

Device masking can also manage access for host platforms that are not supported by this release. If a host can log on to the DMX arrays using a Fibre Channel interface, its access can be controlled.

However, device masking cannot automatically discover the WWNs of host HBAs on unsupported platforms. Instead, you must manually set up a record in the database for these hosts.

Device masking controls overview

SYMCLI commands `symmask and symmaskdb` specifically support the device masking control and monitor operations. With `these` commands, you can define and query the DMX devices that each host's HBA ports are permitted to access.

The `symmask discover` command can be run on both the control station and the managed hosts. The `symmask discover` action locates paths to the device masking database (VCMDB) and assigns alias names (AWWN/AISCSI) to the HBAs residing on the host on which the command is run if they are NULL. You can optionally use the `rename` action to generate aliases to be assigned.

For detailed man page descriptions of these commands and their optons, refer to the *EMC Solutions Enabler CLI Command Reference* .

Database device locking

During the execution of the `symmask` or `symmaskdb` commands, the SYMCLI sets a Symmetrix External Lock (SEL) on the array where the device masking database (VCMDB) resides. This lock ensures that only one host can make changes to the database at any one point in time.

If during the processing of a `symmask` or `symmaskdb` command, the host fails, or a Ctrl/C is performed in the middle of the command, the lock might not release and could lock out further needed changes or control actions. If a device masking command is interrupted and the lock is not released, future invocations of a device masking command will display the following error message:

```
The operation failed because another process has an exclusive lock
on the local Symmetrix.
```

To further examine the presence of this lock, use the following form:

```
symcfg -sid SymmID list -lock -lockn ALL
```

⚠CAUTION

Use the release action only if you believe the lock was forgotten and there are no other operations in progress to the database.

The command will list external locks being held. For this case, it will show a number 14 device masking lock and the length of time it has been on.

To release this lock, use the following form:

```
symcfg -sid SymmID -lockn 14 release
```

Supported topologies

Device masking supports Fibre Channel (point-to-point, arbitrated loop, and switch fabric) and native iSCSI topologies.

Fibre Channel topology

Device masking supports two different Fibre Channel network configurations:

- **Point-to-point** — Network A (left) is configured with a direct connections from one HBA on each host to one FA on the array, providing each host with access to a different set of devices.
- **Multi-initiator** — Network B (right) is configured with multiple hosts accessing the same DMX devices through a common fabric.

Native iSCSI topology

In contrast to the Fibre Channel topology, in a native iSCSI environment, hosts are connected to an array through an Ethernet switch.

CHAP

Native iSCSI support standards require that a security protocol be available. Enginuity version 5670 provides the Challenge Handshake Authentication Protocol (CHAP), which can be enabled or disabled by the user. Refer to "iSCSI software driver configuration" on page 350 for more information about CHAP authentication.

HBA initiator support

Both HBA and DMX director ports in the topology are uniquely identified by a name (WWN or iSCSI). For ease of use, you can associate an ASCII nickname (AWWN and AISCSI).

SYMCLI device masking supports the following HBA initiators:

- World Wide Name (*wwn*) and alias for World Wide Name (*awwn*)
- Native iSCSI over TCP/IP (*iscsi*) and alias for iSCSI over TCP/IP (*aiscsi*)

Refer to "Discovering host HBAs" on page 364 for more information about WWN and iSCSI initiators.

Note: An iSCSI initiator can log in to the array (through a Gig-E port) before the VCMDB is initialized. After the database is initialized, only iSCSI initiators with a masking record are allowed to log in. If the database is initialized by a host connected to the array through a Gig-E port, a record for the iSCSI initiator of that host will be added to the database so that access to the array is not lost as a result of this command.

Using the device masking database

By default, the device masking database (VCMDB) is accessible to all HBAs that log in to the director port where the database is configured. Thus, any host with access privileges can effectively modify the contents of the database if it has device masking installed.

To prevent an unauthorized host from changing the database, the DMX operating system allows you to control a host's access to the database device through the contents of the database records. Only HBAs with valid records in the database can access the database.

The device masking data base (VCMDB) can be unmapped from any director that is not being used for masking control.

Note: If you have PowerPath installed, you should keep the VCMDB mapped.

Configuring device masking

Before you begin using Solutions Enabler device masking, it is important to understand your device masking and test your scripts in a controlled environment.

When you set up your device masking environment, you should initialize the device masking VCMDB (see "Initializing the database" on page 369). This should only be done on an inital setup, since it clears the device of any and all data.

Once you understand the Solutions Enabler device masking functionality described herein, you can begin to develop device masking scripts customized for your environment.

Access control environment setup

If Symmetrix Access Control is being used to protect DMX devices, the host from which you run the device masking commands must be configured in an access control group with an ACL (Access Control List) granting VLOGIX rights to ALL_DEVS. Otherwise, changes to the configuration records in the VCMDB would fail.

Some controls require DIRCTRL or CFGSYM rights. Initialize, restore, and convert controls require CFGSYM and VLOGIX rights. Authentication and `symconnect` controls require DIRCTRL rights.

Configuring device masking involves four steps:

1. "Identifying configuration components" on page 346
2. "Initializing and updating the database" on page 347
3. "Enabling authentication" on page 347
4. "Recommendations for activating the configuration" on page 363

You should be logged on to the control station as Administrator on a Windows system or as root on a UNIX system.

Identifying configuration components

Discover local HBAs on a host that have a channel to a DMX array and generate an AWWN for any HBA that does not have an AWWN assigned yet, updating the VCMDB with the new information.

```
symmask discover hba
```

Figure 13 shows device masking components to identify.

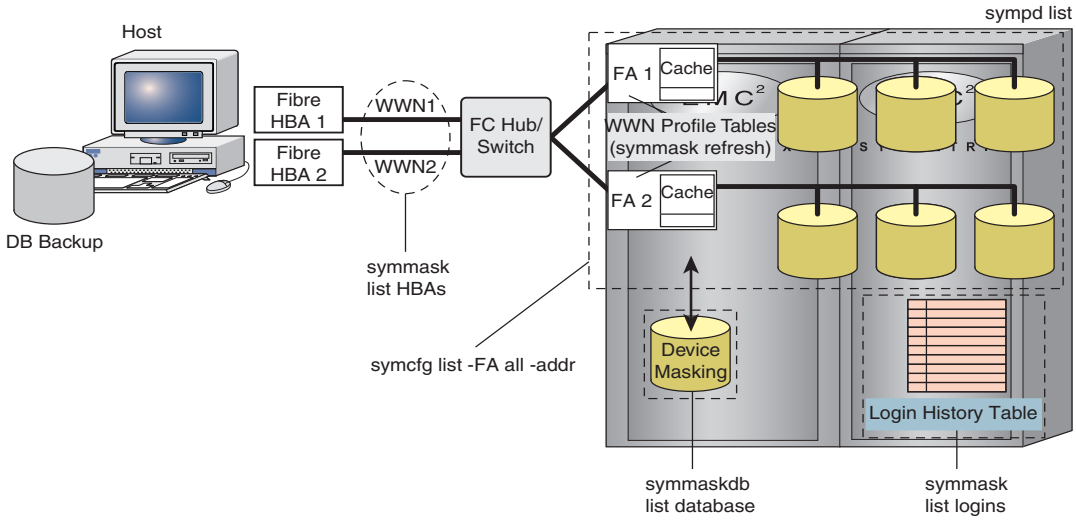


Figure 13 Device masking components

The following commands can be used to obtain the identifiers of each discovered element:

- List the physical device names of all the device masking devices:

  ```
  sympd list -vcm
  ```

- List the HBAs on a host and the FA director port to which each HBA is connected. Note the WWN of the HBA that you will be configuring and which FA port connects to that HBA:

  ```
  symmask list hba
  ```

- DMX director port to which each HBA on the host connects:

  ```
  symmask list logins
  ```

- List devices mapped to the FA director that you will be configuring (for example, director 16A):

  ```
  symcfg list -FA 16A -addr
  ```

SYMCLI device masking supports both World Wide Name (WWN) and native iSCSI (iSCSI) HBA connections.

Initializing and updating the database

After you have identified each element, you can initialize the database and create records. To initialize the database, use the following command:

```
symmaskdb -sid SymmID init -file BackupFilename [-vcmdb_type <3 | 4| 5>]
```

For more information on initializing the VCMDB, refer to "Initializing the database" on page 369; and for more information on supports database types, refer to "VCMDB database types" on page 371.

Make an entry for the HBA-to-FA connection in the VCMDB, specifying devices that the HBA can access.

For example, add a range of devices (0030 through 0034) to the VCMDB on the array (`-sid 814`), specifying the HBA's WWN and the FA director/port that the HBA connects to.

```
symmask -sid 814 -wwn 20000000c920b484 add devs 0030:0034 -dir 16A -p 0
```

Enabling authentication

Before using the device masking SYMCLI commands with iSCSI, you must configure your iSCSI driver software and authentication information.

The iSCSI authentication is negotiated during the HBA login phase and can be implemented using CHAP (Challange Handshake Authentication Protocol).

In addition, Solutions Enabler provides support for a RADIUS (Remote Authentication Dial-In User Service) server for storing authentication information. For details on configuring your RADIUS server, refer to "RADIUS server configuration" on page 362.

iSCSI CHAP authentication: Enginuity version 57*xx*

Beginning with Enginuity version 5771, connection security for iSCSI ports is set with the `symconnect` command. The `symconnect` functionality allows you to set one-way or two-way CHAP authentication between a host HBA and the array.

If a RADIUS server is configured to store the authentication information, you can set the RADIUS server information and enable the array to look there for the authentication data. For more information on the RADIUS server, refer to "RADIUS server configuration" on page 362.

Setting one-way CHAP authentication

With CHAP one-way authentication, the array challenges the host during the initial link negotiation process and expects to receive a valid credential and CHAP secret in response. When challenged, the host transmits a CHAP credential and CHAP secret to the array. The array looks for this credential and CHAP secret in its own CHAP authentication database or on a RADIUS server (if one is set and turned on). Once a positive authentication occurs, the array sends an acceptance message to the host. However, if the array fails to find any record of the credential/secret pair, it sends a rejection message, and the link is closed.

Setting iSCSI authentication for an authenticator requires that you supply the array ID, the iSCSI name, the CHAP credential name (username), and the CHAP secret (password). The following command inserts an entry for the iSCSI initiator

(iqn.2002-06.com.microsoft.host210) into the array CHAP authentication database. The `-cred` and `-secret` parameters specify the authentication data that the host should send in response to a challenge by the array.

```
symconnect -sid 6208 -iscsi iqn.2002-06.com.microsoft.host210
   set chap -cred MyCredentials -secret MySecret
```

The CHAP protocol secret value (`MySecret` in this case) is a user-defined string up to 32 ASCII characters, or 64 binary characters (binary values should be prefixed with the string `0x`) for UNIX users. Windows users need to specify a secret between 12 and 16 characters and a credential name string between 8 and 256 characters.

The `set chap` action automatically enables CHAP authentication for the iSCSI initiator. You can use `disable chap` to disable CHAP authentication and `enable chap` to turn it back on.

For example:

```
symconnect -sid 6208 -iscsi iqn.2002-06.com.microsoft.host210
   disable chap

symconnect -sid 6208 -iscsi iqn.2002-06.com.microsoft.host210
   enable chap
```

Use the `delete chap` command to disable authentication, as follows:

```
symconnect -sid 6208 -iscsi iqn.2002-06.com.microsoft.host210
   delete chap
```

Setting two-way CHAP authentication

With two-way CHAP authentication, the host challenges and authenticates the array too. Thus, both the host and array act as authenticators, and both act as peers. Configuring two-way authentication requires that you configure a one-way authentication for each communication direction.

"Setting one-way CHAP authentication" on page 347 shows how to set up the array as an authenticator, which covers one direction of communication. This section describes how to configure authentication from the opposite direction.

The following command illustrates how to set up the array as a peer. In this case you must define what credential and secret the array will send when challenged by the host HBA authenticator:

```
symconnect -sid 6208 -dir 2D -p 0 set chap -cred MyPeerCredentials
   -secret MyPeerSecret
```

The `-dir` and `-p` parameters specify the director (`2D`) and port (`0`) on the array (`6208`) through which a host HBA is connected. Make a note of the `-secret` value set here. This value will have to be set in the iSCSI Initiator properties window as described in "Configuring iSCSI for two-way chap authentication" on page 357.

Note: The host authenticator must have a user authentication database that contains matching authentication data for the authentication to be successful.

The `set chap` command automatically enables CHAP authentication for the array director and port.

You can use `disable chap` to disable CHAP authentication and `enable chap` to turn it back on.

For example:

```
symconnect -sid 6208 –dir 2D –p 0 disable chap
symconnect -sid 6208 –dir 2D –p 0 enable chap
```

Use the `disable chap` command to disable authentication, as follows:

```
symconnect -sid 6208 –dir 2D –p 0 delete chap
```

Initializing an authentication database

In most cases, you do not want to clear the data of an existing authentication database. If you are unsure whether an authentication database currently exists for an array, issue a command that displays any existing array CHAP or RADIUS authentication database for the array. For example, to view any authentication database if it exists on array `6208`:

```
symconnect -sid 6208 –list
```

To initialize and clear an authentication database, you must specify a backup file name to safeguard against clearing data in the database that should not be lost. For example, the following commands create backup files for CHAP and RADIUS authentication databases and attempt to write any current authentication data to these backup files prior to initializing:

```
symconnect -sid 6208 init chap –file MyChapInitBackup
symconnect -sid 6208 init radius –file MyRadiusInitBackup
```

Displaying authentication information

You can use the basic `symconnect list` command to display an array's authentication database (for example, any CHAP or RADIUS authentication database for array `20`):

For example:

```
symconnect –sid 20 list
```

To display just the CHAP authentication database, include the `chap` option.

CHAP credentials have been defined in the authentication database for two iSCSI initiators, making it possible for the array to authenticate either iSCSI HBA logging into its port. Credentials have also been defined for the director and port (`SE-2D:0`) in the event that two-way authentication is required and the array needs to submit authentication data to the host.

To display only a RADIUS authentication database, include the `radius` parameter.

Backing up and restoring authentication information

Backing up authentication information on a regular basis ensures that you can restore a good version of the authentication database in case incorrect changes or other abnormalities occur in the current database.

The following command creates a backup version of the Symmetrix user authentication database in a file called `MyChapBackup` (each time you back up the database, you must specify a new file name):

```
symconnect -sid 6208 –file MyChapBackup backup chap
```

The following command restores this authentication database using the backup version in the file `MyChapBackup`:

```
symconnect -sid 6208 -file MyChapBackup restore chap
```

Similarly, you can back up and restore the authentication database on a RADIUS server by specifying `backup radius` and `restore radius`.

To display the contents of a backup authentication file, use the `symconnect list` command with the filename:

```
symconnect list -file MyChapBackup
```

iSCSI software driver configuration

This section provides the requirements and configuration steps for preparing a host system with a native iSCSI initiator and the VCMDB (residing in a DMX array) to communicate over the Microsoft iSCSI software driver and the EMC Multi-Protocol Channel Director. The following is an overview of the configuration process:

1. Collect information about the host computer and the DMX Multi-Protocol Channel director:
   a. Get the iSCSI name and IP address of the director from the array.
   b. Get the iSCSI name of the initiator from the host computer.
2. Configure information in the VCMDB of the array that will allow the host computer to access the DMX devices desired:
   a. Add access to the devices from the host initiator by the iSCSI name to the VCMDB.
   b. Add the iSCSI authentication information (if any) about the host initiator.
   c. Refresh the database.
3. Update the iSCSI initiator with the DMX information:
   a. Establish a target connection using iSCSI between the host computer and the designated director port with/without authentication.
   b. Have the host computer logon to the array over iSCSI and establish target devices on the host computer that will persist through a reboot.
4. Perform the following disk administration on the devices on the array, if needed:
   a. Format
   b. Write signatures
   c. Assign drive letters

Configuration of the iSCSI driver can be completed with CHAP authentication as described in "Configuring iSCSI with CHAP authentication" on page 351, or without authentication as described in "Configuring iSCSI without CHAP authentication" on page 358. Refer to the procedure that meets your environment's authentication requirements.

Requirements

For detailed interoperability information, please refer to E-Lab Interoperability Navigator which can be reached at http://elabnavigator.EMC.com.

Note: Your systems must be installed and connected to the network before beginning the configuration.

Installing the iSCSI software initiator

From the Microsoft download site, download and install Microsoft iSCSI Software Initiator. The setup installs an icon on your desktop.

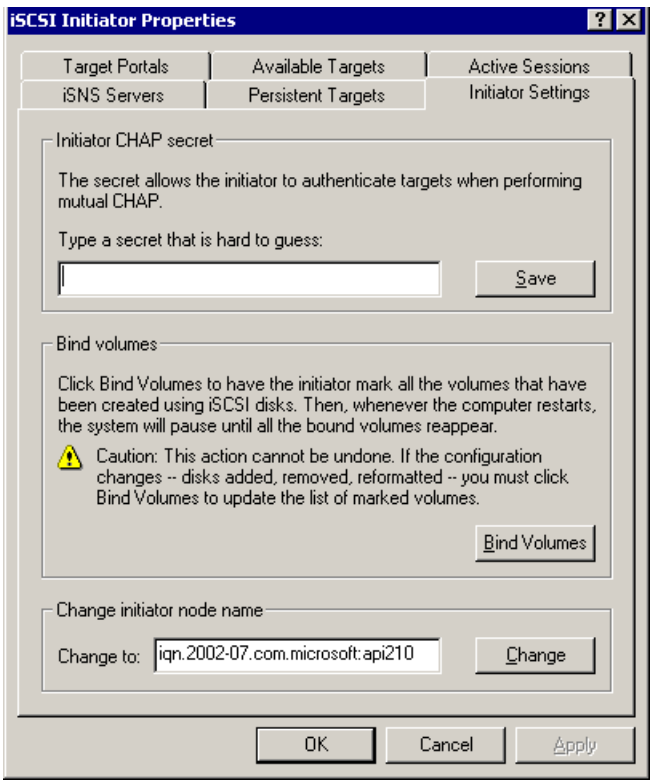
Configuring iSCSI with CHAP authentication

To complete this configuration, you must obtain the iSCSI name of the host, and the iSCSI ID of the multi-protocol director.

Note: This procedure contains steps that require the use of the iSCSI Initiator window and the DOS command window, as follows.

Collect information about the host computer and director

1. Execute the iSCSI Initiator control panel on the host system.
2. Click the **Initiator Settings** tab.



3. Copy the iSCSI ID of your host from the **Change to** field.

   Note: Do not enter any data in this window.

4. From the command line, display and copy the IP address of the multi-protocol director, as shown in this example:

   ```
   symcfg -sid 6208 -dir 3a list -v
   ```

The last two lines of the display contain the iSCSI name and IP address. For example:

```
Symmetrix ID: 000000006208
Product Model                        : DMX2000P
Symmetrix ID                         : 000000006208
Microcode Version (Number)           : 5670 (16260000)
.
iSCSI NAME         : iqn.1992-04.com.emc.5006048000061002
iSCSI IP Address   : 10.10.10.21
```

Configure information in the VCMDB

5. Add a device to create a record in the VCMDB, using the following form:

```
symmask -sid SymmID -iscsi iscsi -dir # -p # add dev #
```

Where:

`SymmID` — Identifies the array ID.
`iscsi` — Specifies the iSCSI name (from step 3).
`-dir` *#* — Identifies the array director number.
`-p` *#* — Identifies the array port number.
`add dev` *#* — Specifies the array device number(s).

For example:

```
symmask -sid 6208 -iscsi iqn.2002-07.com.microsoft:api210 -dir 3a -p 0
   add dev 0023
```

6. Set the CHAP authentication in the VCMDB using the SYMCLI, which is done differently for each supported Enginuity version:

- For Enginuity version **5771** or higher:

```
symconnect -sid SymmID -iscsi iscsi
   set chap -cred Credential -secret Secret
```

Where:

`SymmID` — Identifies the DMX array ID.
`iscsi` — Specifies the host iSCSI name.
`Credential` — Uses between 8 and 256 alphanumeric characters.
`Secret` — UNIX: 32 ASCII characters, or 64 binary characters (binary values should be prefixed with the string `0x`).
Windows: 12 and 16 ASCII characters.

- For Enginuity version **5670** or higher:

```
symmask -sid SymmID -iscsi iscsi
   set authentication -type CHAP
   -credential Credential -secret Secret
```

Where:

`SymmID` — Identifies the DMX array ID.
`iscsi` — Specifies the host iSCSI name.
`CHAP` — Indicates the authentication type.
`CHAPcredential` — Uses 8 alphanumeric characters.
`CHAPsecret` — Uses 12-16 alphanumeric characters.

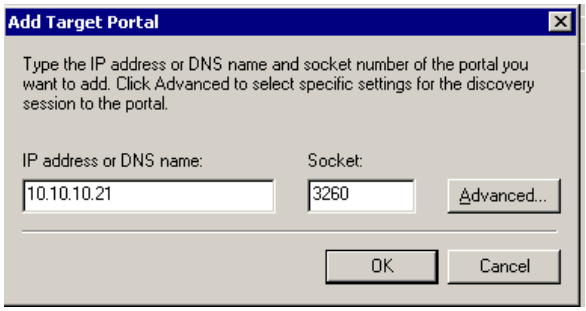
Note: For details about setting iSCSI Authentication, refer to "iSCSI CHAP authentication: Enginuity version 57xx" on page 347 or "iSCSI CHAP authentication: Enginuity version 57xx" on page 347.

7. Refresh the VCMDB, as shown in the following example:

```
symmask -sid 6208 refresh
```

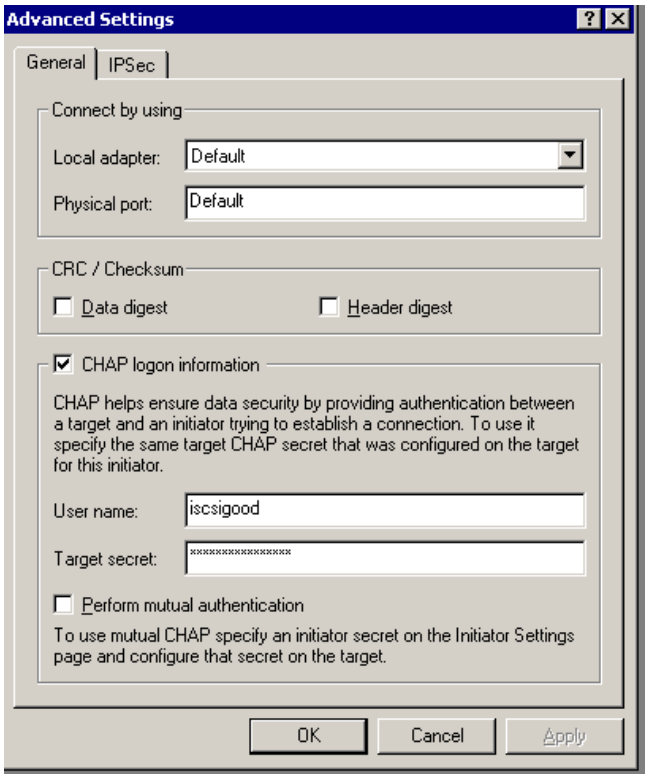
Update iSCSI initiator with DMX information

8. From the iSCSI Initiator Properties window, click the **Target Portals** tab and click **Add**. The **Add Target Portal** dialog box appears.



9. Enter the iSCSI IP address (from step 4) of the multi-protocol director in the **IP address or DNS name** box and click **Advanced**.

   The **Advanced Settings** dialog box appears. The iSCSI name automatically displays in the **User name** field. Do not use this for your credential.



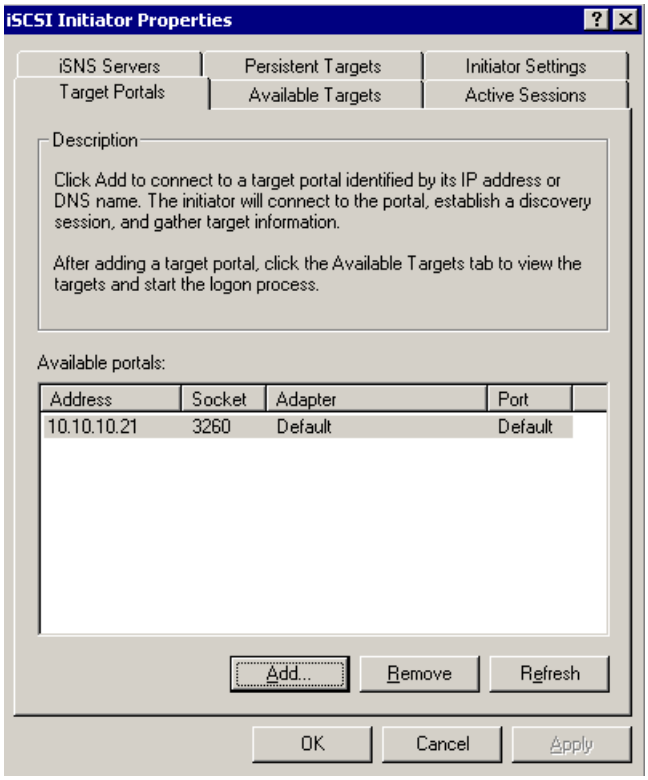
10. Check the box labeled **CHAP logon information**. Change the CHAP credential in the **User name** field, and enter a CHAP secret in the **Target secret** box. Click **OK**.

    The credential name string must be between 8 and 256 characters. The CHAP protocol secret value on UNIX can be 32 ASCII characters, or 64 binary characters (binary values should be prefixed with the string 0x). On Windows the secret must be between 12 and 16 ASCII characters.

    Note: Do not check or change anything else in this dialog box.

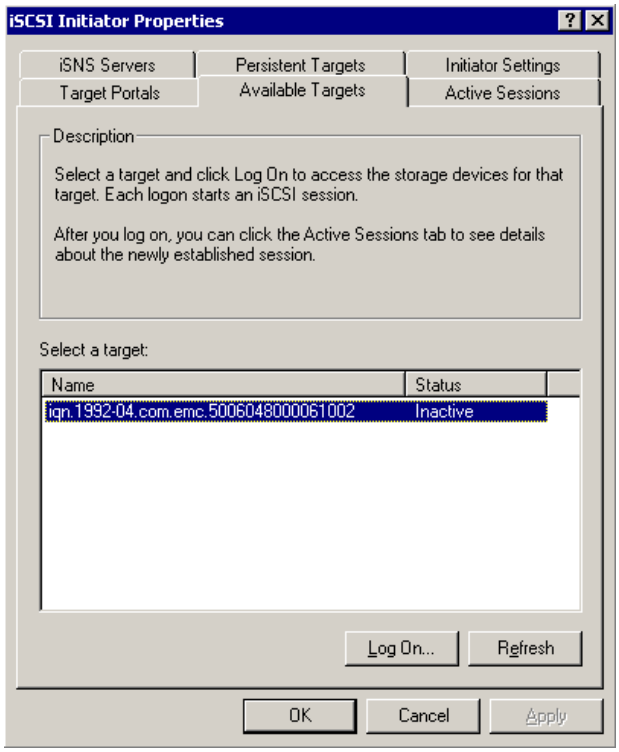
11. Click **OK** to close the **Advanced Settings** dialog box, and click **OK** to close the **Target Portal** dialog box.

    The multi-protocol director IP address should appear in the **Available portals** list in the **Target Portals** window.

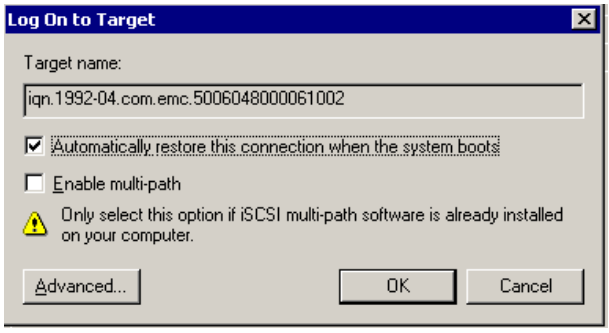


    Note: If an error displays, select the IP address from the **Available portals** list and click **Remove**. Begin the configuration procedure again.

12. Click the **Available Targets** tab. The iSCSI name of the multi-protocol director (from step 4) displays in the **Select a target** list.



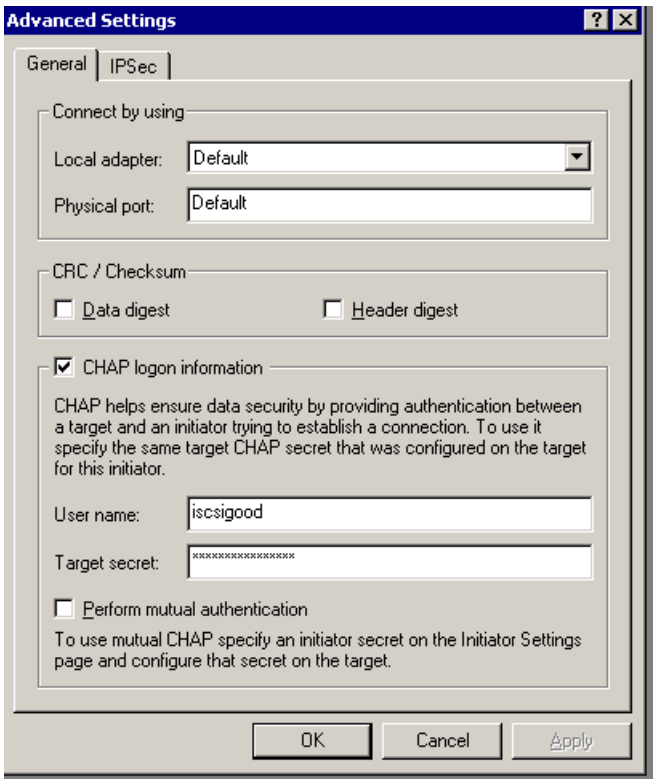
13. Click **Log On**. The **Log On to Target** window dialog box appears.



14. Check **Automatically restore this connection when the system boots,** and click **Advanced**.

    The **Advanced Settings** dialog box appears. The iSCSI name automatically displays in the **User name** field.

    Do not use this for your credential.

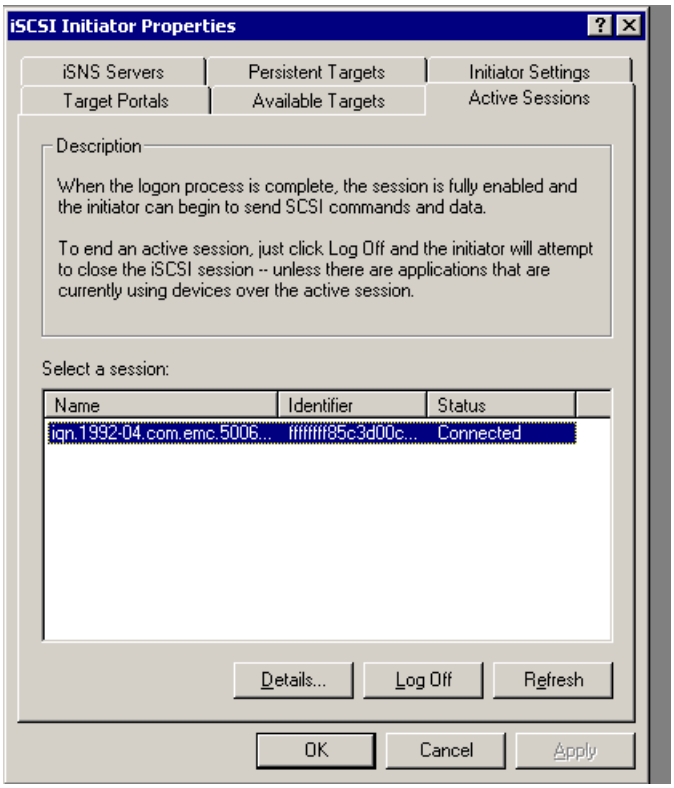


15. Select the box labeled **CHAP logon information**. Change the CHAP credential in the **User name** field, and enter a CHAP secret in the **Target secret** box. Click **OK**.

    The credential name string must be between 8 and 256 characters. The CHAP protocol secret value on UNIX can be 32 ASCII characters, or 64 binary characters (binary values should be prefixed with the string 0x). On Windows the secret value must be between 12 and 16 ASCII characters.

    Note: Do not select or change anything else in this window.

16. From the iSCSI Initiator Properties window, click the **Active Sessions** tab. An active session should display in the **Select a session** list.



17. Select the session and click **Details** to display the disks found by the iSCSI driver.
18. Click **OK** to exit from the iSCSI Initiator.

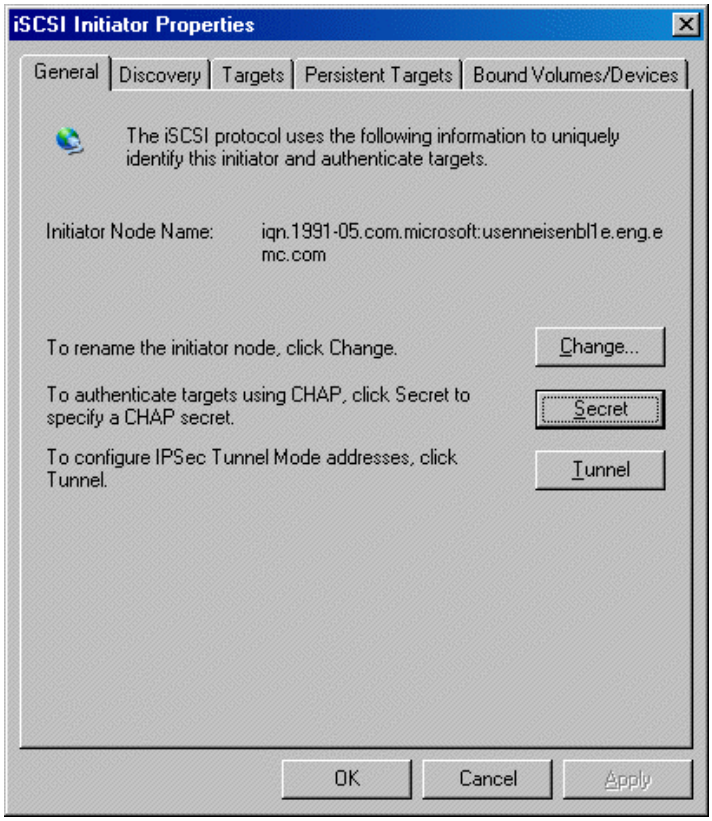
Perform disk administration on the DMX devices

19. Perform any necessary disk administration, such as, formatting, write signatures, and assigning drive letters.
20. Reboot the host system.

Note: The configuration changes you made will not take effect until you reboot your host.

Configuring iSCSI for two-way chap authentication

For a two-way CHAP configuration, after installing your iSCSI Initiator, open your iSCSI Initiator properties, click the **Secret** button on the **General** tab, and set the secret for two-way CHAP authentication to match the value set using `symconnect` as described in "Setting two-way CHAP authentication" on page 348.



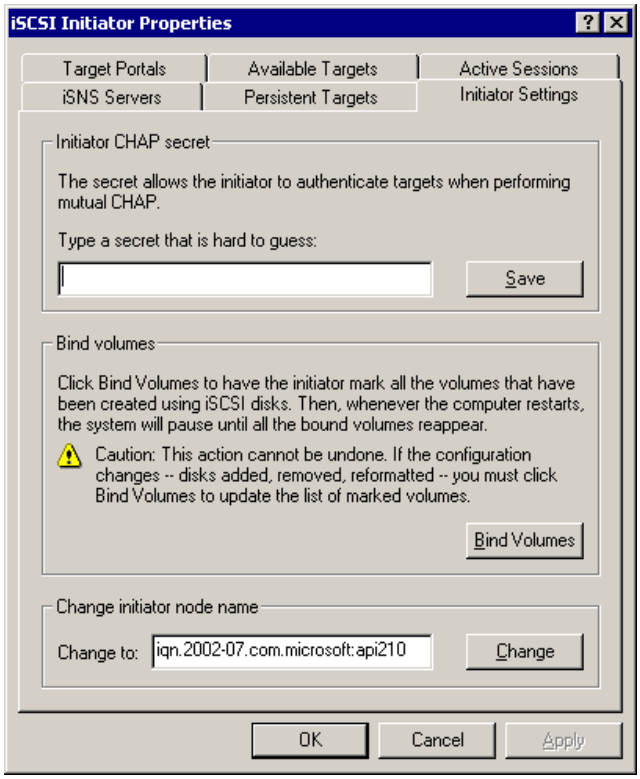
Configuring iSCSI without CHAP authentication

To complete this configuration, you must obtain the iSCSI name of the host, and the iSCSI ID of the multi-protocol director.

Note: This procedure contains steps that require the use of the iSCSI Initiator Properties window and the SYMCLI command window, as follows.

Collect information about the host computer and DMX director

1. Open the iSCSI Initiator on the host system.
2. Click the **Initiator Settings** tab.



3. Copy the iSCSI ID of your host from the **Change to** field at the bottom of the window.

   Note: Do not enter any data in this window.

4. List the iSCSI name and IP address of the multi-protocol director, as shown in this example:

   `symcfg -sid 6208 -dir 3a list -v`

   The last two lines of the display contain the iSCSI name and IP address. For example:

```
Symmetrix ID: 000000006208
Product Model                        : DMX2000P
Symmetrix ID                         : 000000006208
Microcode Version (Number)           : 5670 (16260000)
.
iSCSI NAME          : iqn.1992-04.com.emc.5006048000061002
iSCSI IP Address    : 10.10.10.21
```

Configure information in the VCMDB

5. From the command line, add a device to create a record in the VCMDB, using the following form:

   `symmask -sid` *`SymmID`* `-iscsi` *`iscsi`* `-dir # -p # add dev #`

Where:

`SymmID` — The DMX array ID.

`iscsi` — The iSCSI name copied in step 3.

`-dir` *#* — DMX director number.

`-p` *#* — DMX port number.

`add dev #` — DMX device number.

For example:

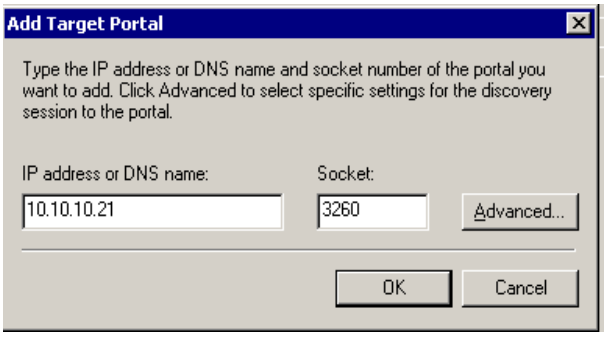
```
symmask -sid 6208 -iscsi iqn.2002-07.com.microsoft:api210 -dir 3a -p 0
      add dev 0023
```

6. Refresh the VCMDB, as shown in the following example:

```
symmask -sid 6208 refresh
```

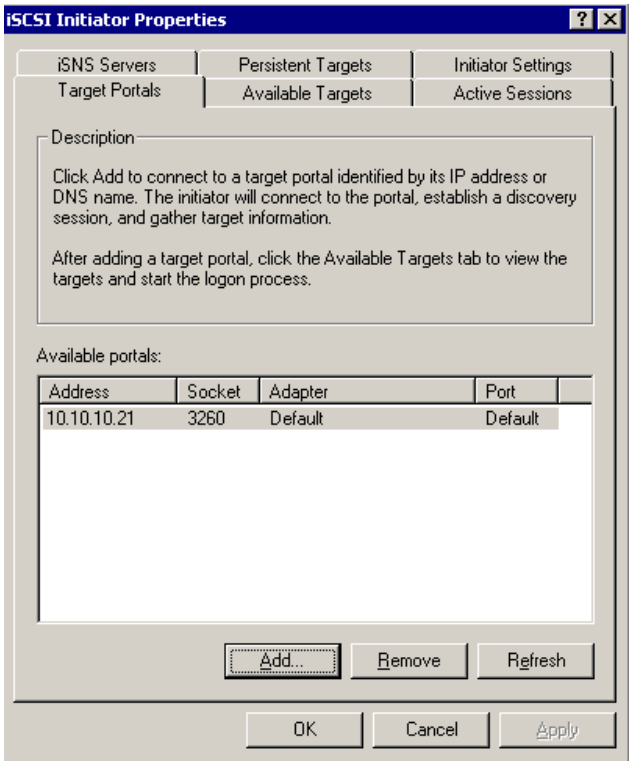
7. From the iSCSI Initiator window, select the **Target Portals** tab and click **Add**.

   The **Add Target Portal** dialog box appears.



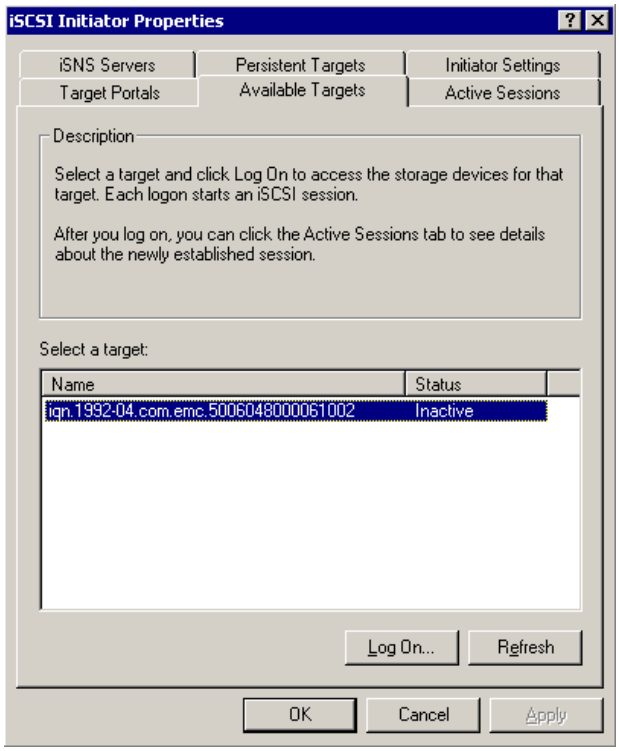
8. Enter the iSCSI IP address (from step 4) of the multi-protocol director in the **IP address or DNS name** box and click **OK**.

   The multi-protocol director IP address should appear in the **Available portals** list in the **Target Portals** window.

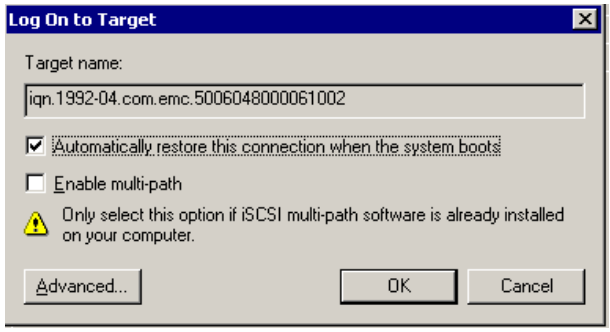


Note: If an error displays, select the IP address from the **Available portals** list and click **Remove**. Begin the configuration procedure again.

9. Click the **Available Targets** tab. The iSCSI name of the multi-protocol director (from step 4) appears in the **Select a target** list.

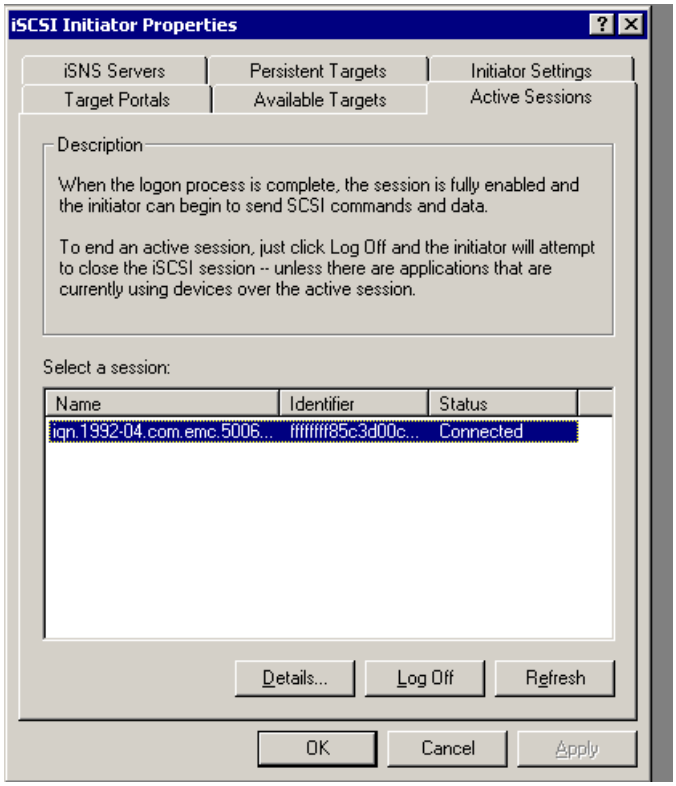


10. Click **Log On**. The **Log On to Target** dialog box appears.



11. Select **Automatically restore this connection when the system boots**, and click **OK**.

12. Click the **Active Sessions** tab. An active session should display in the **Select a session** list.



13. Select the session and click **Details** to display the disks found by the iSCSI driver.
14. Click **OK** to exit from the **iSCSI Initiator Properties** window.

Perform disk administration on the DMX devices

15. Perform any necessary disk administration, such as, formatting, write signatures, and assigning drive letters.
16. Reboot the host system.

Note: The configuration changes you made will not take effect until you reboot your host.

RADIUS server configuration

When using a RADIUS server to store authentication data, you need to use the `symconnect` command to make the SYMAPI server aware of your RADIUS server. Once registered, the SYMAPI server will look to the RADIUS server for matching authentication data when attempting to authenticate a host's credential and secret.

You set the RADIUS server information at the director/port level, letting the array know where to go for the authentication information when a host HBA tries to log in. The required information includes the server name, its IP address, its key (a password of up to 256 characters), and its rank (`primary`, `backup1`, or `backup2`). For example:

```
symconnect -sid 6208 -dir 2D -p 0 set radius -server AP22
   -key MyServerPassword -ip 108.15.139.220 -rank primary
```

The `-rank` parameter establishes server priority preference if two or three RADIUS servers are specified for a given director-port combination. If a primary and two backup servers are enabled, the system tries them in the obvious order until successful (a server might be unreachable for some reason). If primary is disabled, the system looks to `backup1` and `backup2`.

If the RADIUS server port to be used is different than the server's default port, you can specify it using the `-port` option. You can also include options that specify a server retry interval in seconds (`-i`) and the number of times to retry (`-c`).

The `set radius` command automatically enables RADIUS authentication for the array's director and port. You can use `disable radius` to disable authentication and `enable radius` to turn it back on:

For example:

```
symconnect -sid 6208 -dir 2D -p 0 disable radius -rank primary

symconnect -sid 6208 -dir 2D -p 0 enable radius -rank primary
```

Once you have set and enabled all RADIUS server information and at the director and port level, you can then enable the server authentication feature for a given iSCSI HBA logging into a port. Use either `set radius` or `enable radius` to turn on the RADIUS server authentication for the specified iSCSI initiator.

For example:

```
symconnect -sid 6208 -iscsi iqn.2002-06.com.microsoft.host210
   enable radius
```

Similarly, you can use `disable radius` to turn off RADIUS authentication for the iSCSI initiator.

For example:

Use `delete radius` to delete the RADIUS authentication entry, as follows:

```
symconnect -sid 6208 -iscsi iqn.2002-06.com.microsoft.host210
   delete radius
```

Recommendations for activating the configuration

To ensure that updates to the VCMDB become active and visible to your host, it is recommended that you:

1. Back up the device masking VCMDB to a file by calling `symmaskdb backup`.

   Note: During the initial setup you cannot create a backup.

2. Update the DMX array with the configuration changes by performing `symmask refresh`. This calls the DMX director to refresh its WWN/iSCSI-related profile tables in cache with the contents of the device masking VCMDB.

CAUTION

Before running the `symmask refresh` command, or the `symmask -verify` command, make sure there are no HBAs accessing devices in the masked channel (applications running or user activity).

3. Reboot all hosts that have had devices added or removed for the changes to take effect.

   Note: When you reboot a host, you must run `symcfg discover` to scan the DMX devices and refresh the SYMAPI configuration database.

4. When configuration of the database is complete, use `sympd list` to view the DMX devices that can be seen by the host.

Discovering host HBAs

During the initial setup, an administrator runs `symmask discover` on the controlling host to search the environment for DMX devices on each HBA by using the following command:

```
symmask discover hba
```

Note: It is assumed that the host from which these commands are run has access to the DMX devices.

When the `symmask discover` finds a host HBA, it reads the login history table and performs the following:

1. Checks whether an alias exists in the device masking VCMDB. If one does, this command writes it to the login history table.
2. If there is no alias in the device masking VCMDB record, or the login history table, it creates an ASCII alias and writes it to the login history table.

   Note: There is a `-rename` option that can be used with this command to force the discovered hostname/HBA name (or IP address) to be written to the login history table and the device masking VCMDB. This will overwrite any existing AWWN/AISCSI record you have previously established.

3. Prints the initiator identifier (WWN/iSCSI) of the HBAs that are connected to the masked channel and DMX array.
4. If the `-rename` option was used with the `symmask discover` command, the initiator identifier and its ASCII alias are written to the device masking VCMDB.

The `symmask discover` command sends information about this connection back to its host system. The `discover` command is the primary mechanism by which hosts other than the control station can learn about their VCMDB paths to the DMX array. This is displayed using the `symmask list hba` command.

Using alias names

Whether you have defined alias names (AWWN/AISCSI) for the various HBAs, or choose to use those assigned by the SYMAPI server during discovery, alias names can be used in the command line, replacing the cumbersome numeric identifiers.

These names, which are stored in the DMX array's login history table, identify the HBAs connected to the network interface. Alias names can be shorter in length and much more recognizable than the cryptic WWNs/iSCSIs.

ASCII format

All alias names (auto-generated or user-defined) have two parts separated by a slash (/), such as *ALIAS*/*ALIAS*. An ASCII alias names generated by the `discover` action consists of two parts: the name of the host and the name of the HBA.

- For Fibre configurations, the adapter number takes the form of the WWN or iSCSI to guarantee uniqueness. For example, the AWWN for a host whose TCP/IP hostname is `john4554b`, on adapter `10000000c920cf87`, would be `john4554b/10000000c920cf87`.
- For Native iSCSI configurations, the values are hostname/IP address.

For more information on renaming Identifiers, refer to "Managing HBA initiators" on page 374.

Renaming identifiers

When using various `symmask` actions (such as adding or removing devices in the device mask) you can target an HBA path by specifying an AWWN or AISCSI in the command line. Once you have established the aliases with the identifiers in the history table and database, you can rename existing aliases with `symmask rename` action using the following form:

```
symmask -sid SymmID -iscsi iscsi rename aiscsiNew
```

For example, you are working with array 0128 and you want to change your HBA of 20000000c920b484 to Solar2b, enter:

```
symmask -sid 0128 -wwn 20000000c920b484 rename Solar2b/b4
```

You can run `symmask list logins` to display the contents of the login history table to examine the existing alias names on a specified array.

Adding masked devices

The DMX devices you want to isolate can be assigned to a specified masked channel (HBA to director port). You can add or remove devices from these masked channels using their DMX device name, or device group name.

If an alias is set in the login history table, the first record for the initiator in the database can be added using that alias. In addition, any attribute can be set by alias name if the alias can be found in the login history table.

Adding devices

To add devices to a specified HBA/director-port channel, use the following form:

```
symmask -sid SymmID -wwn wwn | -awwn awwn |
           -iscsi iscsi | -aiscsi aiscsi | -host HostName
           -dir <#|all> -p <#|all>  [-celerra] [-rp]
           [-reserve_id ResvID[,ResvID[,ResvID]]]

add devs SymDevStart:SymDevEnd |
         <SymDevName | SymDevName,SymDevName,SymDevName...>
             [-dynamic_lun | -lun Addr | <Addr,Addr...>]
           [-noprompt] [-remove_unmapped] [-map]
```

For example, to add devices 0014 and 0015 on array 0128 for access to all directors and ports on Host3b, enter:

```
symmask -sid 0128 -host Host3b -dir all -p all add devs 0014,0015
```

If the devices are not addressed to the specified FA, a warning message displays. If the devices are already assigned in the database to any WWN, an informational prompt displays. To turn off this functionality, use the `-noprompt` option.

Use the `-map` option to additionally map the devices while adding them to the masking database.

Adding metadevices

To add metadevices, add only the *SymDevname* of the device that is designated as the metahead.

Adding reserved devices

If you are using devices that have been reserved, you must supply the device reservation ID.

For example, to add reserved device 0014 on array 0128 for access to Host3b using director 16a, port 0, enter:

```
symmask -sid 0128 -host Host3b -dir 16a -p 0 add dev 0014 -reserve_id 5
```

"Managing device reservations" on page 248 provides more information about device reservations.

Adding device group devices

To add device group devices to a specified HBA/director-port channel, use the following form:

```
add -g DgName [-std] [-bcv] [-vdev] [-tgt] [-noprompt]
           [-dynamic_lun | -lun Addr] [-remove_unmapped]
```

Use the `-std` option to mask only the standard devices; `-bcv` to mask only the BCV devices; `-vdev` to mask only the virtual devices; and `-tgt` to mask only the target devices.

For example, to add the standard devices of device group `prodB` on array 0128 for access to Host3b using director 16a, port 0, enter:

```
symmask -sid 0128 -awwn Host3b/4a add -g prodB -std -dir 16a -p 0
```

If no option is specified, only standard devices (`-std`) will be added.

Note: Device group names are not stored, only the device information is stored.

Adding devices from a device file

You can perform add masking operations on devices in a device file. When creating the device file, enter the source device first, a space, then the target device. Device files can contain comment lines that begin with the pound sign (#). For example:

```
#Device list mask2 for masking
0014 009A
0151 014C
....
```

A Symmetrix ID is required for this option. In addition you can use the keywords `src` and `tgt` to specify that the action should only use the source devices, or only use the target devices.

For example, to mask only the source devices in the `Mask2` file on array `0845`, to director FA-2A and port 0, enter:

```
symmask -sid 0845 -wwn 50060482d5f003c1 -dir FA-2A -p 0 add -f mask2 src
```

If the `src` or `tgt` keywords are not specified, only the source devices (`src`) are acted upon.

Adding devices with dynamic LUN addressing

When adding a list of devices, a device group, or a device range, you can specify the starting LUN address for each device using the `-lun` option, or have SYMAPI assign the LUN address using the `-dynamic_lun` option.

To add a device or devices to a specified HBA/director-port channel, specifying the LUN address use the following form:

```
symmask -sid SymmID -wwn wwn|-awwn awwn|-iscsi iscsi |-aiscsi aiscsi

add devs SymDevStart:SymDevEnd|SymDevName|SymDevName,,,...\
-lun Addr | Addr, Addr, Addr...
-dir # -p # [-noprompt][-dynamic_lun] [-remove_unmapped] [-map]
```

Where:

`-lun` — Specifies starting LUN addresses. You can specify a single starting LUN or multiple LUNs to match the given ranges.

For example:

```
symmask add devs 15,18,20 -lun 0 -wwn 20000000c920b484 -dir 14C -p 1 -sid 12
```

`-dynamic_lun` — Specifies the use of dynamic LUN addressing, however the application assigns the addresses based on what is already in use for the host HBA.

`-removed_unmapped` — Allows you to remove any unmapped devices when adding devices with dynamic addressing. This option converts an existing device masking record to the new dynamic feature in a single step. Records that contain unmapped devices cannot be converted until those devices are removed.

`-map` — Allows you to map devices while adding them to the masking database. This option takes the next available LUN, which may or may not match the masking LUN. The `-map` option is only supported on Enginuity 5773.

For example, to add devices `100` to `105` on array 104 for access to wwn `20000000c920b484` starting at LUN addresses `0`, and using the next available LUN numbers, for director `2d`, port `0`, enter:

```
symmask add devs 100:105 -lun 0 -wwn 20000000c920b484 -dir 2d -p 0 -sid 104
```

When adding devices from a device file or device group, add the `-dynamic_lun` option to the existing syntax for the system to assign the LUN addresses.

For example:

```
symmask add -f C:\my_device_file -dynamic_lun -wwn 20000000c920b484 -dir 3c -p 0 -sid 90
symmask add devs 2C,2E,30 -dynamic_lun -wwn 20000000c920b484 -dir 2a -p 1 -sid 215
```

Removing masked devices

You can remove some of the devices associated with an HBA entry, or you can remove the entire set of devices associated with an HBA entry. Removing some devices (but not all) requires syntax similar to that for adding devices and the same refresh, backup, and discover steps after completing the remove operation. You can remove devices from a masked channel at any time. To remove a device or devices from a masked channel, use the following form:

```
symmask -sid SymmID -wwn wwn|-awwn awwn|-iscsi iscsi|-aiscsi aiscsi
   -name <name>> | -host HostName -dir <#|all> -p <#|all>
[-celerra] [-reserve_id ResvID [,ResvID[,ResvID]]]

   remove devs SymDevStart:SymDevEnd | SymDevName |
   SymDevName,SymDevName,SymDevName... [-unmap] [-force]
```

The `-unmap` option allows you to unmap the devices while removing them from the masking database, if the devices are not masked to any other initiator on that port. This option is only supported for Enginuity 5773.150.

The `-force` option may be useful when you want to quickly remove a range of specified device names that might span device names not part of the existing noncontiguous masked channel. It may also be needed when working with metadevices.

After each set of changes, activate the configuration by performing a refresh (`symmask refresh`), back up the database (`symmaskdb backup`), and reboot the affected hosts.

Note: When ever you reboot a host, you must run `symcfg discover` on all the DMX devices and refresh the SYMAPI database.

For example, to remove devices 0031 and 0033 from the 0030-to-0034 range of devices that was added previously:

```
symmask -sid 814 -wwn 20000000c920b484 remove devs 0031,0033 -dir 16A -p 0
```

Removing metamembers

For example, to remove metamembers from the device masking VCMDB, but keep the metaheads in place, use the following form:

```
symmaskdb -sid SymmID -meta_member remove
```

The `-force` option may be necessary when working with metamembers.

Removing device group devices

To remove device group devices from a specified HBA/director-port channel, use the following form:

```
symmask -sid SymmID -wwn wwn|-awwn awwn|-iscsi iscsi |-aiscsi aiscsi
remove -g DgName [-std] [-bcv] [-vdev] [-force] -dir # -p # [-noprompt]
```

Use the `-std` option to remove only the standard devices; `-bcv` to remove only the BCV devices; and `-vdev` to remove only the virtual devices.

For example, to remove the standard devices of device group `prodB` on array 0128 from access to Host3b using director 16a, port 0, enter:

```
symmask -sid 0128 -awwn Host3b/4a remove -g prodB -std -dir 16a -p 0
```

If no option is specified, only standard devices (`-std`) will be removed.

Note: Device group names are not stored, only the device information is stored.

Removing devices using a device file

For example, to remove the masking entries for only the source devices in the `Mask2` file on array `0845`, to director FA-2A and port 0, enter:

```
symmask -sid 0845 -wwn 50060482d5f003c1 -dir FA-2A -p 0 remove -f mask2 src
```

If the `src` or `tgt` keywords are not specified, only the source devices are acted upon.

Maintaining the device masking database

After the initial setup, records are added, updated, and deleted from the device masking VCMDB each time a SYMCLI command is issued to add or remove devices in a masked channel, a WWN /iSCSI alias is renamed, one HBA is swapped for another, or an HBA is cleared, removing its device mask allocation.

Initializing the database

For the initial setup of any device masking environment, a device reserved (VCM state enabled) in the DMX array must be initialized and formatted for use as the device masking database, better known as the VCMDB.

The initialization clears the disk device of any current data in the process of formatting the database. In most cases, you do not want to clear the data of an existing VCMDB. If you are unsure whether a VCMDB currently exists, issue the command that displays an existing VCMDB.

For example, to view the VCMDB on array 0128:

```
symmaskdb -sid 0128 list database
```

⚠CAUTION

The following command is rarely used. Be sure you want to zero out the device masking VCMDB before proceeding.

To initialize and clear the database device, use the following form:

```
symmaskdb -sid SymmID init -file BackupFilename [-vcmdb_type <3 | 4| 5>]
```

For a safeguard, you must specify a backup filename, since this command will try to write the data from this device to a backup file on your host before it clears the current data.

For example, to initialize the database and create backup file `BackupDevMask1` on array `0128`, enter:

```
symmaskdb -sid 0128 init -file BackupDevMask1
```

If you do not include the `-vcmdb_type` option, the default database type depends on the size of the VCMDB device.

Syncing masking data

Syncing (refreshing) of masking data is done through the symcfg sync command using the -masking qualifier.

For example, to sync masking data on array `0128`, enter:

```
symcfg sync -sid 0128 -masking
```

Preventing unauthorized modification of the VCMDB

By default, the VCMDB grants access to all HBAs that log in to the FA director port where the database is configured. Without preventive measures, any host with access privileges can modify the VCMDB if it has the EMC Solutions Enabler Device Masking component.

Managing a backup VCMDB file

You can create a backup file containing the current contents of the device masking VCMDB. This is useful when you want to temporarily change the access rights or device masking assignments to various HBAs. Then at some point in time, you can return the device masking environment back to the original masked environment. Often, just backing up the database on a regular basis ensures you can recover your established masked environment in the event of some improper changes or failure.

Note: You cannot reuse any existing backup filename. Always create a new file.

To create a backup database file, use the following form:

```
symmaskdb -sid SymmID backup -file BackupFilename
```

For example, to create backup file BackupDevMask on array 0128, enter:

```
symmaskdb -sid 0128 backup -file BackupDevMask
```

The VCMDB backup files vary in length, depending on their contents.

VCMDB database types

Device masking database types define the number of devices per database record and the number of records you can mask.

Beginning with Enginuity 5671, a Type 5 device masking VCMDB is supported, and Enginuity 5771 only supports a Type 6 VCMDB. The various VCMDB types are defined as:

- **Type 3** — Supports up to 8 K devices per record and 32 fibre/32 iSCSI initiator records per port (the VCMDB device must be 24 cylinders or larger)
- **Type 4** — Supports up to 8 K devices per record and 64 fibre/128 iSCSI initiator records per port (the VCMDB device must be 48 cylinders or larger)
- **Type 5** — Supports up to 16 K devices per record and 64 fibre/128 iSCSI initiator records per port (the VCMDB device must be 96 cylinders or larger)
- **Type 6** — Supports up to 64 K devices per record and 256 fibre/512 iSCSI initiator records per port (cannot be specified with SYMCLI)

When initializing the VCMDB database where no database currently exists, SYMCLI defaults to creating a database according the size of the VCMDB device being initialized, For example, a Type 4 VCMDB would be created for a 48-cylinder device. This default behavior is also true for the VCMDB device that currently holds a database (for example, a Type 4 database on a 96-cylinder device will be initialized as a Type 5 database). A Type 4 or 5 database initializes with direct I/O writes blocked to protect against outside sources corrupting the database. You can also block direct I/O writes to a Type 3 database using `set no_direct_io` option.

You can explicitly create a VCMDB of a specific type in the initialization procedure. This is especially important when you are considering backwards compatibility issues.

The following command provides an example syntax specifying a VCMDB type:

```
symmaskdb -sid 814 init -file MyInitBackup -vcmdb_type 4
```

Converting a VCMDB type

You can convert a Type 3 database to Type 4 or Type 5, and a Type 4 to a Type 5, if the size of the VCMDB device on the specified array is large enough for the database type.

The following is an example converts the VCMDB on array 814 to a Type 5 database:

```
symmaskdb -sid 814 convert -vcmdb_type 5 -file MyCvrtBU
```

If you convert from a lower type database to a higher type, any hosts running a Solutions Enabler version that does not support the higher VCMDB type will not be able to access the database. For example, if one host running Solutions Enabler version 6.0 converts an existing VCMDB to Type 5, another connected host running Solutions Enabler version 5.4 will no longer have access to the database until this host is upgraded to version 6.0 or higher.

Restoring a backup VCMDB

For example, you can restore the database from the backup file stored on the host, as follows:

```
symmaskdb -sid 0128 restore -file MyCvrtBU
```

The database is restored as is; Type 3 restores to Type 3, and Type 4 restores to Type 4. You can use the `convert` or `set` options to alter the resulting environments.

For example, to restore the database from a backup file, but not the authentication information, enter:

```
symmaskdb -sid 0128 restore -file MyCvrtBU -skip_authentication
```

Note: The `-skip_authentication` option is not available with Enginuity 5771 and higher.

Restoring a backup and converting its type

To restore from a backup file and convert its database type in a single command specify the `-vcm_type` option and specify the type: 4 or 5.

For example, enter:

```
symmaskdb -sid 0128 restore -file MyCvrtBU -vcm_type 5
```

Blocking direct writes to the VCMDB

If you have a Type 3 database and would like to block direct writes to the database, use the following form:

```
symmaskdb -sid SymmID set -no_direct_io
```

Note: Type 4 and 5 databases initialize with direct I/O writes blocked by default.

Viewing the VCMDB

Once you have initialized your database and created records, you can view various information about the database. In addition, you can view the login history details.

Viewing the database

You can examine the entire contents of the device masking VCMDB in its entirety, by director and port, by WWN name (or alias), or by iSCSI name (or alias) using the following form:

```
symmaskdb -sid SymmID | -file Filename [-v]
[-dir all [-p all] | -dir # [-p <#|all>]]
[-wwn wwn | -awwn awwn |
-iscsi iscsi | -aiscsi aiscsi]
list database
```

Viewing entire VCMDB

For example, to view the device masking VCMDB on array `6196`, enter:

```
symmaskdb -sid 6196 list database
```

Viewing VCMDB records for a director

For example, to examine the database for records concerning director `2b`, port `1` on array `6196`, enter:

```
symmaskdb -sid 6196 list database -dir 2b -p 1
```

You can examine the masked assignment of devices to a specific HBA using the following form:

```
symmaskdb -sid SymmID list devs [-wwn wwn | -awwn awwn|-iscsi iscsi |
-aiscsi aiscsi]
```

Note: The initiator is optional.

For example, to examine the devices on array 343 without supplying a specific initiator, enter:

```
symmaskdb -sid 343 list devs
```

Viewing device capacity

You can view the capacity of devices assigned to a particular host with the following command:

```
symmaskdb -sid SymmID list capacity -host HostName
```

When a hostname is specified, such as api145, SYMAPI looks for records with the AWWN value of api145/*‹wwn›* and pull that record for the device information. If the default AWWN alias has been modified to something other than the host name, this command does not include information for that HBA, or may appear incomplete if only 2 out 3 HBAs for a given host have the hostname in the alias.

Note: The `list database` and `list devs` commands can be targeted to a backup database file on your host by replacing the `-sid` option with a `-file` option that specifies your backup filename.

For example, to view the capacity of host `api145` on array `6196`, enter:

```
symmaskdb -sid 6196 list capacity -host api145
```

This command requires that the first part of the HBA alias be the host name.

Viewing HBA assignments

For example, you can view which HBAs have been assigned to specific devices with the following command:

```
symmaskdb -sid 6196 list assignment -devs 0040:0043
```

Use the `list no_assignment` command to list devices that are mapped, but not yet assigned, in the device masking VCMDB.

Viewing the login history table

The `symmask list logins` command is used to view the login history table. This table in the DMX array lists which hosts and HBAs are logged on to an array for all directors and their director ports. You can also use options to restrict this data to a specific director and/or port number.

For example to return login information for array `6196`, enter:

```
symmask -sid 6196 list logins
```

The following is sample output from this command:

```
Symmetrix ID            : 000000006196

Director Identification : FA-2A
Director Port           : 1

                            User-generated                      Logged On
Identifier       Type  Node Name        Port Name        FCID   In     Fabric
---------------- ----- --------------------------------- ------ ------ -----
10000000c9238053 Fibre api145           i@1f,4000,@2     260e13 Yes    Yes
5006048000060d21 Fibre NULL             NULL             261e13 No     Yes
```

The identifier field indicates which HBA is communicating with the array. User-generated node and port names are identified as the AWWN or AISCSI alias associated with it. Columns labeled On Fabric and Logged In indicate whether the HBA is connected to a fabric and whether it is logged in to the DMX system.

You can use the verbose (`-v`) option to view the last active login information.

Refreshing the database

The device masking records are located in two places on the array: the database and the local directors. When a record is written to the array, it is stored in the database and then pushed out to the directors at a later time. A `symmask refresh` command will trigger all records within the database to be pushed out to the local directors. A login by an HBA only triggers the record for that specific HBA to be updated.

To view what is currently stored on the local directors to see what (if any) affect a `symmask refresh` command will have, use the `-verify` option, as shown in the following example:

```
symmask refresh -verify -sid 0237
```

The command output is similar to the `list database -v` output, with the following differences:

- Includes the state of the VCM flag on the port.
- The first column contains either a minus (-) or a plus (+) when information within the records do not match, or when the record is located in only one location and will be completely added or removed.
- Attributes, such as LUN offset, which are not set in the record(s) are not displayed.

Managing HBA initiators

An HBA can be specified in symmask commands by its unique WWN, by an AWWN alias associated with the WWN, or by an iSCSI name (or its alias). These names appear in the DMX array login history table and in the VCMDB.

Note: Fiber HBAs have a host WWN and a port WWN for each port on the HBA. Most HBAs have only one port, but some have two. In rare cases, an HBA has four ports. HBA identification in device masking always refers to a port WWN.

When you issue the `symmask discover hba` command to update the login history table, the Symmetrix API (SYMAPI) checks the VCMDB to determine whether an AWWN exists for each WWN record. If not, SYMAPI creates an AWWN that consists of two parts (the name of the host and the name of the HBA) and writes it to the login history table. However, you can rename the AWWN to a shorter name, if you prefer.

For example, the following command allows you to examine AWWNs in the login history table of array 814:

```
symmask -sid 814 list logins
```

Creating and renaming aliases

To assign an AWWN to fit your naming requirements, you can use the `symmask rename` command at any time—even before SYMAPI generates an AWWN.

For example, the following command assigns the two-part name Solaris3A/b4 as the AWWN for WWN 20000000c920b484:

```
symmask -sid 814 -wwn 20000000c920b484 rename Solaris3A/b4
```

If later you decide you want to replace all user-defined AWWNs (like Solaris3A/b4) with system-generated AWWNs, the following command overwrites all existing AWWN entries in the VCMDB:

```
symmask discover hba -rename
```

You can identify the HBA in `symmask` commands through an iSCSI name. The iSCSI is a SCSI-over-IP protocol that tries to take advantage of the IP network for SCSI traffic rather than requiring a new Fibre network. An iSCSI name is used like an IP address or a WWN and is displayed using the `symmask list logins` or the `symmask list hba` commands. A unique iSCSI name is determined by the hardware that logs into a DMX array and, like the AWWN, a two-part alias name is generated from the `symmask discover hba` command.

Swapping a new HBA for a failed HBA

In the event a host adapter fails, or needs replacement for any reason, you can replace the adapter and assign its set of devices to a new adapter by using the `replace` action in the following form:

```
symmask -sid SymmID -wwn wwn|-awwn awwn|-iscsi iscsi|-aiscsi aiscsi
replace wwnNew | iscsiNew
```

To swap HBAs:

1. Run `symmask list logins` to view the old WWN/iSCSI HBAs.
2. Swap HBA boards.
3. Run `symmask list hba` or `discover` to view the new initiator (for example WWN).
4. Run `symmask replace` to substitute a new WWN for all occurrences in the database of the old WWN.
5. For example, to replace old WWN 20000000c920b484 with new WWN 20000000c920b393:

   ```
   symmask -sid 814 -wwn 20000000c920b484 replace 20000000c920b393
   ```

6. Run `symmask discover -rename` or `symmask rename` to establish the new AWWN and assign an AWWN to the new HBA in both the VCMDB and the login history table.
7. Run `symmask refresh` to update the director profile tables (in cache) from the database.

Deleting HBA associations

You can also delete (in the database) the set of devices associated to a host adapter by using the `symmask delete` action with the following form:

```
symmask -sid SymmID delete -wwn wwn|-awwn awwn|-iscsi iscsi|-aiscsi aiscsi [-login]
```

For this database record deletion, you can restrict the action to just devices on a specific DMX director and port with the following option:

```
[-dir #|all -p #|all]
```

In addition, on arrays running Enginuity 5671 and higher, you can use the `-login` option to delete the entry from the login history table as well.

To delete a set of devices associated to a host adapter, it is suggested to:

1. Run `symmask delete` action.
2. Update the array with the configuration change by performing a `symmask refresh`. This calls the DMX director to refresh its WWN/iSCSI-related profile tables in cache with the contents of the device masking VCMDB.

⚠CAUTION

Before running the `symmask refresh` or the `symmask refresh -verify` commands, make sure there are no HBAs accessing devices in the masked channel (applications running or user activity).

Managing the Fibre Channel-to-host interface

Using the device masking commands, you can adjust the protocol characteristics of the Fibre Channel-to-host interface to be compatible with your host platform-specific requirements. For your specific host communication protocol, the `symmask set` command allows an *advanced* user to adjust the following attributes on a host adapter port basis, or all ports by using `-dir ALL` / `-p ALL`:

- Fibre Channel ID (FCID) lock down
- Device LUN visibility
- LUN base/offset skip
- Heterogeneous host configuration

A record for the host adapter port assignment must already exist in the VCMDB for these channel attributes to be set.

CAUTION

Do not proceed with any of these adjustments unless you are comfortable with your understanding of the details of your HBA interfaces. Improper settings can disable the use of your host with the array.

Locking down a Fibre Channel ID

Fibre Channel ID (FCID) lockdown is a security feature that limits host device access by adding Fibre Channel ID information of a switch within a fabric to device access records in the device masking VCMDB. This feature handles WWN spoofing and the threat it poses to your networked systems in a shared (same director port) storage port configuration.

For example, to implement the Fibre Channel ID lockdown feature on Fibre Channel 021300 for director 16A, port 0, enter:

```
symmask -sid 018 set lockdown on 021300 -awwn SolarB/1f,0,fca@1,0 -dir 16A -p 0
```

This feature lets you set the Fibre Channel ID (FCID) of the WWN of the HBA you want to protect. The FCID is then added to the database record for the WWN of the specified HBA with the specified director and is locked. Once a Fibre Channel ID is locked, no user with a spoofed WWN can log in. If a user with a spoofed WWN is already logged in, that user loses all access through that HBA.

CAUTION

When an HBA logs into a director port, the Fibre Channel ID accompanies it, telling the director port where to send its response. By specifying Fibre Channel ID information of the switch (in addition to the WWN of the HBA in the device masking record), the valid physical path through the SAN for a particular HBA is locked down. Only an HBA with a Fibre Channel ID that matches the FCID specified in the device masking record is able to log in to the storage port. If the incorrect Fibre Channel ID is added to the device masking VCMDB, that HBA will lose access and the host utilities may hang on the server with the locked out WWN. It is recommended that at least two HBAs be available on the administrator host. If one HBA becomes locked out, the host will have access through the other HBA and can correct the record in the database.

Lockdown steps

To find the Fibre Channel ID, lock it down, verify that it is locked down, and then force the change to take effect, use the following procedure:

1. Find the WWN. If the device for the device masking VCMDB is visible, run `symmask list hba` to find the device path of the HBA you want to protect.

   Note: If the VCMDB is unmapped, no PDEVs will be visible when the `symmask list hba` command is issued. The `sympd` command must be called to set an alternate path.

2. Find the Fibre Channel ID value by using one of the following methods:

   - Run `symmask list logins -pdev`, specifying the device path you found in step 1, to find the Fibre Channel ID of the WWN of the HBA you want to protect.

- Find the Fibre Channel ID value on the switch, refer to "Finding the FCID of a switch" on page 378.

3. Run `symmask set lockdown` set to on with the FCID of the Fibre Channel ID you found in step 2.
4. Run `symmaskdb list database` in verbose mode (`-v`) to verify that the Fibre Channel ID is locked down.
5. Either reboot the host or pull the cable from the director and then replace the cable. This causes the change to take effect. If you reboot, you must run `symcfg discover` to refresh the SYMAPI database.

Effects on other commands

This section describes how locking down a Fibre Channel ID affects other commands:

- `symmask delete` — Locking down a Fibre Channel ID has no effect on the `delete` action. The specified record is completely cleared from the database.
- `symmask replace` — Locking down a Fibre Channel ID has no effect on the `replace` action when the cable is simply moved from one HBA to another and not moved at the switch. In this case, the Fibre Channel ID value that is already in place in the database remains the same for the new HBA.

  However, if the cable is moved from one port on the switch to another, the FCID value changes. Do not unlock the Fibre Channel ID during this swap. Instead, leave at least one path open to the database device, and reset the FCID value after the swap by recalling the `set` action.

Since you do not have a path from the HBA whose Fibre Channel ID you want to lock down, you cannot use `symmask list logins` to find the FCID value. Instead you must obtain the FCID value from the switch.

Finding the FCID of a switch

This section describes how to find the Fibre Channel ID on Connectrix® and Brocade switches:

- Connectrix switch — Through the hardware view, click the board and then the port of the switch whose Fibre Channel ID you want to find. Right-click to display the port properties window that includes the FCID value.
- Brocade switch — Telnet to the switch and run `nsShow`. Look for the PID value of the WWN of the HBA you want to protect, which is the Fibre Channel ID value.

Format of a FCID

The Fibre Channel ID basically incorporates the port and the domain ID of the switch in the fabric into which the HBA is plugged.

For example:

Connectrix ED-1032 and Brocade 1000 series:

2 <u>2</u> ***0 4*** 1 3

<u>Underlined text</u> is the domain. ***Bold Italic text*** is the port.

In this example, the domain is 2 and the port is 04.

Note: For Connectrix, the port is offset by 4.

Brocade 2000 series DS-16B:

0 2 1 ***3*** 0 0

Underlined text is the domain. ***Bold Italic text*** is the port.

In this example, the domain is 02 and the port is 3.

Setting device LUN visibility

The device LUN visibility feature allows the host driver to discover devices with noncontiguous LUN addresses. During the process of discovery, the host operating system scans for LUNs starting at 000 and continuing to a point where it does not find a LUN in the sequence. If there is no LUN 000 on the target director, or there is a break in the sequence of LUNs on that target, some operating systems (notably HP-UX and Linux) do not detect the remaining LUNs and fail to discover noncontiguous devices.

To allow your host to detect all these devices, turn on the device LUN visibility feature so that all devices attached to a specified FA director and port are made available to the HBA.

For example, to turn on visibility for FA director 16A, port 0, when working with a host HBA whose WWN is 20000000c920b484:

```
symmask -sid 814 set visibility on -dir 16A -p 0 -wwn 20000000c920b484
```

Allowing the host to scan the other devices past a break in the LUN sequence does not change how the host accesses them. For example, if an HP-UX host has LUNs 0000, 0002, 0003, 0004 assigned to it, that is exactly what the host will see. If you need the host to access this same sequence without the break between 0000 and 0002, you can adjust host visibility by offsetting LUN addresses as described in "Setting the LUN base/offset skip adjustment" on page 379.

Setting the LUN base/offset skip adjustment

Certain host platforms require LUN 000 to be present when it scans the interface for devices. For this feature, LUN refers to the SCSI LUN number. Also, these host types and others cannot see devices beyond the initial contiguous LUN sequence (they cannot skip over masked holes in an array of intended devices). In a device masking environment, this can be a problem when you need to mask out certain devices from the visibility range of certain host platforms.

For these host platforms, the device masking LUN base/offset skip adjustment feature provides the ability to specify a LUN *base* and an *offset* hexidecimal value for the skip hole (recorded in the database). When the host asks for a LUN that is equal to, or greater than the skip hole base value, the offset is added to the LUN value requested by the host to render the actual LUN (device) in the array. The *base* value is essentially the host's first missing LUN in the skip hole. The offset is the hole size (number of addresses needed to skip over the hole).

To set LUN base and offset values for a skip hole within an HBA to director channel, use the following form:

```
symmask -sid SymmID set lunoffset on offset base -awwn awwn -dir # -p #
```

For example, (with director 16A/port 0) to make LUNs (devices) 005 through 008 available to host HPB03/1, you need a LUN base address of 000 and an offset of 5 (to skip over 000-004):

```
symmask -sid 018 set lunoffset on 5 0 -awwn HPB03/1 -dir 16A -p 0
```

Multiple hosts and broken sequences

On the other hand, if your masked devices for an HBA-to-FA connection had LUN addresses 0000–0003 and 0007–0009, you would need to specify a LUN base address of 0004 and an offset of 0003 (to renumber LUNs 0007–0009 as LUNs 0004–0006). You can record only one gap per HBA-to-FA connection.

If you have multiple hosts that cannot discover devices with noncontiguous LUN addresses, you need to issue the `symmask set lunoffset` command for each host.

As shown in Table 23 for Scenario 1, you could have LUN devices 000 through 006 assigned to Host A and 007 through 00A assigned to Host B. Because in this case, Host B needs to see LUN 000 first, you would have to `set lunoffset on` with a base value of 000 and an offset of 7. For this case, Host A does not have a problem since there is no hole in its assigned device sequence and it starts with 000.

Table 23 LUN base/offset scenarios for multiple hosts with skip holes

	Host A LUNs	Host B LUNs	base	offset
Scenario 1	000-006		-	-
		007-00A	000	7
Scenario 2	000-002		-	-
	007-008		003	4
		003-006	000	3

For Scenario 2, you could have LUNs 000 through 002 and 007 through 008 assigned to Host A. Host B could have LUNs 003 through 006. Host A's 000-002 is not a problem, but LUNs 007 -008 require a skip hole base value of 003 (because the first visible sequence stopped at 002) and an offset of 4 (hole size). Also, Host B's LUNs 003-006 requires a skip base value of 000 and an offset of 3. This scenario would require two commands: one targeting Host A and one targeting Host B. Only one skip hole per HBA channel can be recorded in the database.

Setting the HBA port flags

HBA port flags can be set on a per initiator basis or by host name. This feature allows specific host flags to be enabled and disabled on a specified director and port or on all directors and all ports. Setting a port flag by host name or for all ports requires Enginuity 5773.150 and higher.

Note: Setting HBA port flags replaces setting the heterogeneous host configuration flags in the next section. The heterogeneous host configuration types listed in Table 24 on page 382 continue to be valid, but will not be expanded. To switch to setting HBA port flags, the heterogeneous host configuration must be disabled for a given director/port/HBA and all flags must be reset.

To set (or reset) the HBA flags, use the following form:

```
symmask -sid SymmID -wwn wwn | -awwn awwn |
        -iscsi iscsi | -aiscsi aiscsi | -host HostName

set hba_flags
           <on <<flag>,<flag>...> <-enable | -disable> |
            off [<flag>,<flag>...] >
           -dir <#|all> -p <#|all>
```

Where:

`hba_flags` — Sets the record in the database to hold information on the HBA port setting that may differ than the current setting on the corresponding FA.

`on | off` — Turns HBA flags on or off.

`flag` — Specifies the overrided HBA port flags from the values [in brackets]:

Common_Serial_Number [C]	AS400 [AS4]
Disable_Q_Reset_on_UA [D]	OpenVMS [OVMS]
Environ_Set [E]	SCSI_3 [SC3]
Siemens [S]	Sunapee [SCL]
Volume_Set_Addressing [V]	Sequent [SEQ]
Avoid_Reset_Broadcast [ARB]	SPC2_Protocol_Version [SPC2]
	SCSI_Support1 [OS2007]

`-enable` — Enables the specified HBA port flag(s) on a per initiator basis.

`-disable` — Disables the specified HBA port flag(s) on a per initiator basis.

For example, to turn on HBA flags and enable the `Common_Serial_Number` for host `api1182` on array `343` for all directors and ports, enter:

```
symmask -sid 343 -host api1182 set hba_flags on C -enable -dir all -p all
```

Setting the heterogeneous host configuration

Note: Setting the heterogeneous host configuration has been superseded by "Setting the HBA port flags". The heterogeneous host configuration types listed in Table 24 continue to be valid, but will not be expanded. To switch to setting HBA port flags, the heterogeneous host configuration must be disabled on the array, and all flags must be reset.

Heterogeneous host configuration is a feature that allows different host types to share a single director FA port even though they may require different port settings for their distinctive interface protocol.

Turning on heterogeneous host, turns on the bits listed in Table 24, (and turns off the bits listed in footnote `a`) on the given director/port (for the given WWN or iSCSI). If this feature is enabled for one host type for a WWN, it must be disabled for that WWN before a new host type can be assigned.

This feature can be used in conjunction with the LUN offset skip feature to allow the different hosts their own LUN addressing scheme. With that scheme, the devices they see are different from those seen by any other host on the director.

The following syntax is used to set certain heterogeneous host configuration flags to optimize the host-to-director interface:

```
symmask -sid SymmID -wwn wwn -dir # -p # set heterogeneous on HostConfigFlag
```

Possible *HostConfigFlag* values are listed in the last column of Table 24.

Table 24 Host platforms and interface configuration flags (page 1 of 2)

Host platform	Requirements	Bit[a]	Host configuration flag (*HostConfigFlag*)
AS/400		AS4	AS400
AS/400	Load source extender	AS4, V	AS400_LSE
Bull Escala/AIX			BULL_AIX
Bull Escala/AIX	PowerPath V1.5.x or earlier	D	BULL_AIX_PP15
Celerra		ARB	CELERRA
HP/DEC AlphaServers Tru64 UNIX 5.x FC-SW		OVMS	DEC_UNIX
HP/DEC OpenVMS		SC3, OVMS	DEC_OVMS
Data General AViiON NUMA 25000 Server		D	DG_AViiON
FSC BS2000/OSD Servers		C, D	FSC_2000
FSC PRIMEPOWER GP7000F Series host			PRIMEPOWER
FSC PRIMEPOWER GP7000F Series host	PowerPath V1.5.x or earlier	D	PRIMEPOWER_PP15
FSC PRIMEPOWER GP7000F Series host	VERITAS DMP	C, D	PRIMEPOWER_DMP
Fujitsu Services ICL Open VME		C	ICL_OPEN
Hewlett-Packard HP-UX		C, V	HP_UX
IBM AIX with FC 6227, 6228, 6239		SC3	IBM_AIX
IBM AIX with FC 6227, 6228, 6239	PowerPath V1.5.x or earlier	SC3, D	IBM_AIX_PP15
IBM AIX with FC 6227, 6228, 6239		C, SC3	IBM_AIX_DMP
IBM AIX with FC 6227, 6228, 6239		C, D, SC3	IBM_AIX_DMP_PP15
IBM AIX with EMC Fibre Channel			IBM_EMC
IBM AIX with EMC Fibre Channel	PowerPath V1.5.x or earlier	D	IBM_EMC_PP15
Linux			LINUX
Linux		C	LINUX_DMP
Linux	Veritas Cluster Server	D, C, SC3	LINUX_DMP_VCS
NCR MP-RAS/Windows NT			NCR
NCR MP-RAS/Windows NT	Multiple vendor platforms	D	NCR_MP
NCR MP-RAS/Windows NT	If Windows NT is used with TNT, set FBA Env. Sense key to 4; otherwise, set it to 6	E	NCR_NT
NCR MP-RAS/Windows NT	Multiple vendor platforms:If Windows NT is used with TNT, set FBA Env. Sense key to 4; otherwise, set it to 6	D, E	NCR_NT_MP

Table 24 Host platforms and interface configuration flags (page 2 of 2)

Host platform	Requirements	Bit[a]	Host configuration flag (*HostConfigFlag*)
Novell NetWare			NOVELL
Novell NetWare	Cluster	D	NOVELL_CLUSTER
Windows NT/Windows 2000			WINDOWS
Windows NT/Windows 2000	PowerPath V1.5.x or earlier	D	WINDOWS_PP15
Windows NT/Windows 2000	HP/Agilent controllers	V	WINDOWS_HP
Windows NT/Windows 2000	PowerPath V1.5.x or earlier, HP/Agilent controllers	D, V	WINDOWS_HP_PP15
Windows NT/Windows 2000	VERITAS VxVM DMP	C	WINDOWS_DMP
Windows NT/Windows 2000	HP/Agilent controllers VERITAS VxVM DMP	C, V	WINDOWS_HP_DMP
Windows NT/Windows 2000	PowerPath V1.5.x or earlier, VERITAS VxVM DMP	C, D	WINDOWS_DMP_PP15
Windows NT/Windows 2000	HP/Agilent and PowerPath V1.5.x, VERITAS VxVM DMP	C, D, V	WINDOWS_HP_DMP_PP15
Sequent NUMA-Q		E, C, SEQ	SEQUENT
Sequent NUMA-Q	FC-SW configurations only	E, C, SEQ, V	SEQUENT_FCSW
FSC Reliant UNIX RM series		E, D, S	RELIANT
Sun			SOLARIS
Sun	PowerPath V1.5.x or earlier	D	SOLARIS_PP15
Sun	VERITAS DMP	C, D	SOLARIS_DMP
Sun	Sun Cluster (ealier than 3.0)	C, D, SCL	SUN_CLUSTER
Sun	Sun Cluster (3.0 or higher)	C	SUN_CLUSTER30
VERITAS Cluster (VCS), EMC GeoSpan for VCS		D	VERITAS
VERITAS Cluster (VCS), EMC GeoSpan for VCS	VERITAS DMP	C, D	VERITAS_DMP
VERITAS Cluster (VCS), EMC GeoSpan for VCS	VSC 2.0 or higher		VERITAS20
VMware (ESX 2.5 or earlier only)		C, SC3*	VMWARE

1. The following defines the host characteristic for each of the bits used in the table:

AS4	AS/400 secondary port
C	Common serial number for multipaths
D	Disable Queue Reset on Unit Attention (UA)
E	Environmental reports to host from DMX
S	Enable Siemens host RM/400 - RM/600

SCL	Enable Sunapee (for Sun PDB clusters)
SC3	SCSI 3 interface
SEQ	Sequent Host (DYNIX/ptx)
OVMS	OpenVMS Fibre connection
V	Enable volume set addressing
ARB	Avoid Reset Broadcast
SPC2	SPC2 Protocol (Enginuity 5x71 and higher)
SC3*	Bit will only be set if Enginuity Version is lower than 5670
0S2007	Enable SCSI_Support1

CHAPTER 12
Managing Quality of Service

This chapter describes how to manage the Quality of Service (QoS) on devices in your storage environment and how to manage dynamic cache partitions in your array using the SYMCLI `symqos` command.

- Overview 386
- Setting copy pace and priority 386
- Setting host I/O priority (Enginuity 5876 or lower) 389
- Setting bandwidth for Open Replicator (Enginuity 5876 or lower) 390
- Setting and resetting LRU parameters 391
- Setting RDF director workload distribution 391
- Managing dynamic cache partitions (Enginuity 5876 or lower) 393

Overview

The QoS (Quality of Service) SYMCLI feature allows you to adjust the data transfer (copy) pace on specified devices, or devices in a device group or storage group, of certain operations.

QoS allows you more flexibility in managing your array's performance. By increasing the response time for specific copy operations on selected devices or groups, you can increase the overall performance of the other array devices.

The `symqos` command provides Quality of Service controls on specified devices. Possible operations are:

- Sets the copy pace for device groups and storage groups.
- Queries the QoS copy priorities for members of a specified group.
- Sets and lists the LRU cache assignments to a device or device group.
- Resets the LRU assignment back to the default LRU.
- Enables the display of the SRDF director CPU workload distributions.
- Creates, modifies, and deletes cache partitions.
- Adds and removes a device or a range of devices to a cache partition.
- Renames and empties cache partitions.
- Enables analysis of cache partitions.
- Lists specific devices in a cache partition.
- Enables/disables workload distribution on RDF directors.
- Sets/resets workload distribution synchronous I/Os, asynchronous I/Os, and copy I/Os.
- Lists the workload distribution settings for RDF directors.
- Sets and lists the maximum allowed bandwidth percentage for Open Replicator usage for a given director, port, director/port, or all directors/ports.
- Enables, disables, and sets priority service time for a range of devices, a device group, or a storage group. (This feature is not supported for VMAX arrays running HYPERMAX OS 5977 or higher.)

Setting copy pace and priority

Using SYMCLI, you can set QoS parameters and values for specified devices managing the copy pace (priority) and the LRU assignments.

Starting with Enginuity 5875, the pace and priority settings are supported for VLUNs and discontinued for BCV devices. In addition, the `symqos` command now supports storage groups as a target parameter.

Copy pace and priority settings

Different copy pace (and priority) levels can be set on certain types of devices. Possible copy pace levels on the devices are 0, the highest, to 16, the lowest. Zero is the default. Use the following form when setting the copy pace.

```
symqos -g DgName [-bcv | -nobcv]
symqos [<-sg SgName |-devs <SymDevStart:SymDevEnd | SymDevName
       [,<SymDevStart:SymDevEnd | SymDevName> . . .]>>]

set <BCV|RDF|MIR|CLONE|VLUN> pace|priority
    <URGENT|Value|STOP> [LdevName]
```

If a device group is not specified, a storage group or devices must be specified. The definitions for the syntax follow.

Where:

`-dg` — Indicates the copy operation is on a device group devices.

`-sg` — Indicates the copy operation is on storage group devices.

`-bcv` – Specifies to only work on BCV group members.

`-nobcv` – Specifies to work on group members that are not BCVs.

`-devs` — Indicates the copy operation is on a range of devices.

`BCV` — TimeFinder copy operations.

Note: BCV is not supported for the URGENT and STOP values.

`RDF` — SRDF copy operations

`MIR` — Mirror copy operations

`CLONE` — Clone (previously called Snap) copy operations

`VLUN` — VLUN copy operations

Note: Setting the pace/priority for VLUN requires Enginuity 5875 or 5876. This option is not available with HYPERMAX OS 5977 or higher.

Unless an *LdevName* is specified, this command defaults to setting parameters on STD and BCV devices.

`URGENT` — The copy pace is classified as urgent and may be faster than the default.

`Value` — An integer value between 0 (fastest pace) to 16 (slowest pace). If the value URGENT is used, the copy pace is classified as urgent, and may be faster than 0, the default value. If the value STOP is used, copying is stopped.

Note: The URGENT and STOP actions are supported with Enginuity 5875 and higher.

`Stop` — The background initiated copying will stop.

Note: The syntax items `pace` and `priority` are interchangeable nouns, which can be used to identify and set the same parameter.

With Enginuity 5875 and higher, the copy is driven from the target DA. This change increases optimized writes, and limits cache consumption, allowing tighter control of copy scheduling. QoS can still be applied on the target or source volumes, unless there are multiple clone sessions on the source device. In that case, applying QoS on the source will apply to all copy sessions. Setting on the target will only apply to that particular session.

You can specify a different clone copy priority for each clone session associated with the same source device. To set a separate clone priority for each session, the `set` operation needs to be performed on the clone target device. When the clone priority is set on the

clone source device, all clone sessions associated with that device have the same clone priority. When priority is set on both the source and target devices, the target priority takes precedence and the source clone priority for that session is ignored.

Copy pacing examples

The `ProdA group` has 1000 volumes and is in the Precopied state, and the `ProdB` group has 10 volume pairs, all of which are also source devices for `ProdA`.

If devices in `ProdB` require copying quickly, set QoS only on the target devices to URGENT. Or alternatively, set QoS on `ProdA` to priority 16 to slow the copy of the 1000 volumes.

For SRDF operations, in device group `ProdDB`, to set the copy pace level to `2` for all STD and BCV devices, enter:

```
symqos set RDF pace 2 -g ProdDB
```

For SRDF operations, to set the SRDF copy pace value to `1` on just clone devices of device group `ProdDB`, enter:

```
symqos set CLONE pace 1 -g ProdDB -bcv
```

To set a copy pace to `1` during copy operations for mirror device `DEV021` in group `ProdDB`, enter:

```
symqos set MIR pace 1 -g ProdDB DEV021
```

Using query

The following command and sample output shows the display for the `symqos query` command for a device group:

```
symqos -g dg_test1 query

Disk Group (DG) Name     : dg_test1
DG's Type                : REGULAR
DG's Symmetrix ID        : 000194900341


Device Name                                          Copy Pace
------------------------------------------- -------------- -------------------
Logical     Physical                     Sym Config         BCV RDF MIR CLN VLN
------------------------------------------- -------------- -------------------
DEV001      Not Visible                00250 Unprotected      0   0   0   0   0
DEV002      Not Visible                00251 Unprotected      0   0   0   0   0
DEV003      Not Visible                00252 Unprotected      0   0   0   0   0
DEV004      Not Visible                00253 Unprotected      0   0   0   0   0
DEV005      Not Visible                00254 Unprotected      0   0   0   0   0
. . .
Legend:
  Copy Pace : STP = Copy is stopped,
              URG = Copy has urgent priority.
```

Listing copy pace settings

You can list copy pace settings for devices, a device group, or a storage group.

- When copy pace is set on a storage group, the output only shows data specific to that storage group.
- When copy pace is set on a device, the array ID displays at the top.

- When copy pace is set on a device group, the output only shows data specific to that device group, as shown in the following example command and output:

Note: For arrays running HYPERMAX OS 5977, the `symqos list` command output displays N/A for VLN copy pace information.

Setting host I/O priority (Enginuity 5876 or lower)

Note: Enabling priority of service is not supported on arrays running HYPERMAX OS 5977. or VMAX 10K arrays. The message "The requested feature is not supported for this Enginuity or SYMAPI version" displays.

Using Symmetrix Priority Controls (SPC), you can use the `symqos` command to enable and prioritize the service time of the host I/O to a device, a range of devices, or devices in a device or storage group. The following device types cannot be assigned a priority:

- DATA
- DLDEV
- DRV
- SAVE
- SFS
- Thin
- VAULT
- VDEV

Enabling and disabling priority of service

To enable or disable the priority of service feature for an array, use the following form:

```
symqos -pst -sid SymmID enable | disable
```

For example, to enable host I/O device priority feature for array `1234`, enter:

```
symqos -pst -sid 1234 enable
```

Setting host I/O priority

When priority of service is enabled, all eligible devices in the array are assigned priority 1 by default. To assign a new priority, use the following form:

```
symqos -pst -sid SymmID
[-devs <SymDevStart:SymDevEnd | SymDevName
          [,<SymDevStart:SymDevEnd | SymDevName>...]>] |
-g DgName] [-std] [-bcv]

set hostio priority Value
```

Where:

`-pst` — Indicates a priority service time operation.

`set hostio priority` — Assigns host I/O priority to a range of devices or devices in a device group. When setting the priority for devices in a device group, the priority can optionally be assigned to STD devices, BCV devices, or both.

`Value` — For host I/O priority setting, values are range from 1 (highest priority) to 16 (lowest priority).

For example, to assign all STD devices of device group `DeviceGroup` to host I/O priority 4, enter:

```
symqos -pst -g DeviceGroup set hostio priority 4 -std
```

If neither STD or BCV is specified, the default is STD.

Listing priority of service

To view the priority of service for a range of devices or devices in a device group, or a given host I/O priority, use the following form:

```
symqos -pst

   list -sid SymmID [-devs <SymDevStart:SymDevEnd | SymDevName
          [,<SymDevStart:SymDevEnd | SymDevName>...]>]
             [-all | -hostio_priority <<Value|ALL>]

        list -g DgName [-std] [-bcv]
             [-hostio_priority <Value|ALL>]
```

For example, to list the priority of service time for array `131`, devices `002` and `003`, enter:

```
symqos -pst -sid 131 list -hostio_priority -devs 002:003
```

When listing device groups, if the device type is not specified, the display will list both STD and BCV devices.

For complete syntax for the `symqos` command, refer to the *EMC Solutions Enabler SYMCLI Command Reference*.

Setting bandwidth for Open Replicator (Enginuity 5876 or lower)

For Solutions Enabler V8.0.1, setting and listing of port throughput is performed using the `symqos` command. (These operations were previously performed with the `symrcopy` command.)

To set (or list) the maximum allowed bandwidth for Open Replicator usage for a given director, port, director and port, or all directors and ports, use the following form:

```
symqos -rcopy -sid SymmID
   -dir <# | ALL> [-p <# | ALL>]

   set ceiling <CeilingPercent | DISABLE | NONE>

   list ceiling
```

Acceptable values for the *`CeilingPercent`* are 0-100, DISABLE or NONE. If the value DISABLE is used, this will block Open Replicator I/O on the director/port specified. If the value NONE is used, this will turn off the ceiling function on this director/port, effectively enabling Open Replicator pace for sessions using this director/port.

Setting and resetting LRU parameters

Using SYMCLI, you can set QoS parameters and values for specified devices managing LRU cache I/O operations.

The contention for cache access can be managed by the least recently used (LRU) ring partitions in the array cache. You can manage the cache strategy utilizing LRU rings (groups) and assigning certain devices or device groups to them. The syntax follows:

```
set LRU <#,#,# | ALL> [LdevName]
set LRUname LRUname [LdevName]

reset LRU [LdevName]
```

To assign all devices of device group `ProdDB` to LRU group `2`, enter:

```
symqos -g ProdDB set LRU 2
```

Or, to assign all devices of device group `ProdDB` to an LRU group named `GROUP_2`, enter:

```
symqos -g ProdDB set LRUname GROUP_2
```

Using the `-lru` option of `symcfg list`, you can list the cache-slot allocation and allocation percentage of a specified LRU cache management group number. You can list all the LRUs if you specify `ALL`.

Using the `reset` option of `symqos`, you can reset the LRU assignment back to the default LRU. For example:

```
symqos reset LRU LdevName
```

Setting RDF director workload distribution

Solutions Enabler supports mixed SRDF workloads on the same RDF director. The supported workloads for this feature are:

- Synchronous I/Os
- Asynchronous I/Os
- Copy I/Os

The configuration can be done on a source or target array, but it only affects the director that contains the R1 devices in the RDF group. You may want to configure both sides the same if your devices ever swap the RDF personality.

Solution Enabler provides an interface to enable and disable the RDF director CPU resource distribution and to display the current setting. If the feature is disabled, then the pre-5876 Enginuity behavior for RDF director CPU resource distribution will be in effect.

Use the `symqos` command to set and display the default RDF director CPU resource distribution assigned to each type of workload. The default RDF director CPU distribution will be used for an RDF director if the feature is enabled and an RDF director specific value has not been set or has been reset.

Use the `symqos` command to set, reset, and display the RDF director CPU resource distribution assigned to each type of workload on a specific RDF director.

Note: Any change made to a specific SRDF director's settings will take effect immediately for that director. Any change made to the system default settings will take effect immediately for all SRDF directors that do not have explicit director level settings.

Feature requirements

This feature requires the following:

- The array must be running Enginuity 5876 or HYPERMAX OS 5977.
- You must have SYMAPI_ACCESS_TYPE_CFGSYM access control privilege to the array involved in the action.
- The array must have an SRDF license.

Enabling and disabling the RDF director workload distribution

Use the following syntax to enable this feature:

```
symqos -RA -sid SymmID

   enable -io
   disable -io
```

To enable the workload percentage settings for Synchronous, Asynchronous, and Copy I/Os on array 1234, enter:

```
symqos -RA -sid 1234 enable -io
```

Setting the RDF director workload distribution

Use the following syntax to set the I/O defaults on the SRDF director:

```
symqos -RA -sid SymmID

   set IO -default
       -sync SyncPercent -async AsyncPercent -copy CopyPercent

   set IO -dir <# | ALL>
       -sync SyncPercent -async AsyncPercent -copy CopyPercent
```

Each percent value is the percentage of RDF director CPU resources for each type of I/Os. These specified percentages should add up to 100.

For example, to set the RDF director workload percentages on array 1234 to 60% for Synchronous I/Os, 30% for Asynchronous I/Os and 10% for Copy I/Os, enter:

```
symqos -RA -sid 1234 set IO -default -sync 60
           -async 30 -copy 10
```

For example, to set the RDF workload percentages on director 8G of array 1234 to 50% for Synchronous I/Os, 30% for Asynchronous I/Os and 20% for Copy I/Os, enter:

```
symqos -RA -sid 1234 -dir 8G set IO -sync 50
           -async 30 -copy 20
```

Resetting the RDF director workload distribution

Use the `reset` command to reset the workload percentages to the array default values:

```
reset IO -dir <# | ALL>
```

For example, to reset the settings of the workload percentages on director 8G of array 1234, enter:

```
symqos -RA -sid 1234 -dir 8G reset IO
```

Workload based re-distribution of RA QoS values

When one or more of the workload types are not active, the QoS values automatically adjust so that the RDF director is always capable of operating at 100%.

Workload types are polled for, and in the absence of one or more workload types, the workload distribution values are automatically re-distributed to the types of SRDF replication that are active.

The following three examples illustrate the redistribution values:

- Example 1 — The default values are set at 70, 20, 10. However there is only SRDF/A and Adaptive copy workload active. Since there is a 2:1 ratio between the SRDF/A and adaptive copy settings, the QoS values automatically adjust to 66.6.for SRDF/A and 33.3 for Adaptive copy.
- Example 2— The default values are set at 70, 20, 10. However, there is only SRDF/S and Adaptive copy workload active. Since there is a 3.5:1 ratio between the SRDF/S and adaptive copy settings, the QoS values automatically adjust to 78 for SRDF/S and 22 for Adaptive copy.
- Example 3 — Custom QOS values have been set at 60, 30, 10. The only active workload is Adaptive copy. The adaptive copy workload will consume 100% of the RDF CPU.

When multiple RDF groups exist on an RDF director, each active RDF group shares an equal percentage of the available CPU resource (within their QoS parameters). In other words, if there were 2 RDF groups running SRDF/S in example 2 above, each would get 39% of the RDF CPU. As workload types start and stop, the RA QoS values adjust accordingly.

To display the RDF director workload distribution, issue the `symqos -RA list -io` command.

Viewing the audit log

To display the generated logs for the `symqos` commands, issue a `symaudit` list -text command.

Managing dynamic cache partitions (Enginuity 5876 or lower)

Note: Cache partitioning is not supported on arrays running HYPERMAX OS 5977.

Dynamic Cache Partitioning divides the cache memory into multiple partitions with unique names and their device path assignments. Partition areas can be made static or dynamic in size. The dynamic partitioning provides flexibility to the amount of floating memory that can be allocated with a high and low watermark. This allows memory resources to be temporarily donated to other partitions when needed. The `symqos` command allows you to create partitions for different device groupings in addition to the default partition that all devices belong to initially. Each partition has a target cache percentage as well as a

minimum and maximum percentage. In addition, you can donate unused cache to other partitions after a a specified donation time. With Enginuity 5876, the number of allowed cache partitions is 16.

Enabling and disabling cache partitions

Cache partitioning is enabled and disabled on the array. To enable or disable cache partitions, use the following form:

```
symqos -cp -sid SymmID enable | disable
```

For example, to enable cache partition settings on array 301, enter:

```
symqos -cp -sid 301 enable
```

Creating cache partitions

Before creating a cache partition, note the following:

- There must be an available partition.
- There must be enough cache left in the default partition that it does not fall below the minimum required cache.

You need to create a cache partition before you can add devices to it. To create a cache partition, use the following form:

```
symqos -cp -name PartitionName -sid SymmID

   create -target TargetPercent
          -min MinimumPercent
          -max MaximumPercent
          -wp WritePendingLimit
          -time DonationTime
```

Where:

`-cp` — Indicates a cache partition operation.

`PartitionName` — Identifies the partition by a unique name. The name can be up to 31 characters.

`TargetPercent` — Specifies the target cache percentage for a cache partition.

`MinimumPercent` — Specifies the minimum cache percentage for a cache partition.

`MaximumPercent` — Specifies the maximum cache percentage for a cache partition.

`WritePendingLimit` — Specifies the write pending limit percentage for a cache partition. Possible values are 40 - 80.

`DonationTime` — Specifies the donation time in seconds; the time for when idle cache is made available to other cache partitions.

For example, to create a cache partition named `TestPartition` on array 1234, enter:

```
symqos -cp -name TestPartition -sid 1234 -target 10 -min 5 -max 40 -wp 50 -time 10 create
```

Adding devices to a cache partition

You can add single devices, a range of devices, or devices in a device group or SRDF group to a cache partition. The `addall` option must be used with device ranges, SRDF groups, or pools. With device groups, you can specify all devices (`addall)`, logical devices (`ld`), standard devices (`-std`), BCV devices (`-bcv`), or virtual devices (`-vdev)`.

Device restrictions

When adding devices to a cache partition:

- Only specify the head device for metadevices and CKD devices.
- Do not add VAULT devices, as these are blocked from being moved to any new partition.
- Ensure that all devices in SRDF groups that are in Async mode belong to a single partition.
- With Enginuity version 5876, if a device range is specified with the `addall` or `rmall` commands, DATA devices are included in the operation. With lower versions of Enginuity, DATA devices are blocked.

To add devices to a partition, use the following form:

```
symqos -cp -name PartitionName -sid SymmID
   add dev SymDevName

symqos -cp -name PartitionName -sid SymmID
     [-devs <SymDevStart:SymDevEnd | SymDevName
          [,<SymDevStart:SymDevEnd | SymDevName>...]>] |
        -rdfg GrpNum | -pool PoolName -snap | -rdfa_dse | -thin]
   addall

symqos -cp -name PartitionName -g DgName
   add ld LdevName
   addall [-std] [-bcv][-vdev]
```

The following examples show how to add different device types to a cache partition. If the device type is not specified, the default is to add only STD devices.

To add device `001` to partition `TestP` on array `310`, enter:

```
symqos -cp -name TestP -sid 310 add dev 001
```

To add device range `002` through `012` to partition `TestP` on array `310`, enter:

```
symqos -cp -name TestP -sid 310 -devs 002:012 addall
```

To add the BCV devices in device group `Finance` to the partition `TestP` on array `310`, enter:

```
-symqos -cp -name TestP -sid 310 -g Finance addall -bcv
```

With Enginuity 5876 and higher, device pools can be added to and removed from cache partitions.

To add a thin pool named `Finance` to the cache partition `TestA`, enter:

```
symqos -cp TestA -sid 310 addall -pool Finance -thin
```

Modifying a cache partition

Use the `modify` argument to change the values of a cache partition, as follows:

```
symqos -cp -name PartitionName -ran-sid SymmID

   modify [-target TargetPercent]
          [-min MinimumPercent]
          [-max MaximumPercent]
          [-wp WritePendingLimit]
          [-time DonationTime]
```

The parameter definitions are listed earlier in "Creating cache partitions" on page 394.

Renaming a cache partition

Use the `rename -new_name` option to rename the cache partition, as follows:

```
symqos -cp -name PartitionName -sid SymmID
   rename -new_name NewPartitionName
```

For example, to rename cache partition `TestP` on array `310` to `TestA`, enter:

```
symqos -cp TestP -sid 310 rename -new_name TestA
```

Deleting a cache partition

Use the `symqos delete` command to delete a cache partition. When deleting a cache partition, all devices, groups, and pools currently assigned to that partition will move back to the default partition.

Removing devices from a cache partition

Use the `remove dev`, `rmall`, and `remove ld` commands to remove devices from a cache partition and return them to the default partition. This can be done for a device range, a device group, SRDF group, or pool.

To remove the BCV devices in device group `Finance` from the partition `TestP` on array `310`, enter:

```
symqos -cp -name TestP -sid 310 -g Finance remall -bcv
```

Using analyze mode

Dynamic Cache Partitioning provides an `analyze` mode, which you can use to determine how much actual cache your applications require. While in `analyze` mode, applications will not fail due to cache partition overflow.

Cache partitions cannot be changed from enabled to `analyze`; the cache partition must be disabled to use the `analyze` mode. The syntax for setting the cache partition to analyze mode follows:

```
symqos -cp -sid SymmID
   analyze
```

To run in `analyze` mode, set the target percent, and then set all partitions with a minimum of 0%, a maximum of 100%, and a donation time of 0 seconds. After running in `analyze` mode, you can display the cache partition settings with the `-usage` option, and determine more accurate cache partition target values.

The `-usage` option is shown in the following sample command and output:

```
symqos -cp list -sid 237 -settings -usage

Symmetrix ID: 000190300237

    Cache Partitioning         : Enabled
    Number of Partitions       : 2
    Max Num of Partitions      : 8
    Min Allowed Target %       : 10
    Max Allowed Target %       : 90
    XRC Partition State        : Enabled
    XRC Partition Name         : XRC_partition
    Empty Partition Removal    : Remove
    Time of Last Modification  : 01:12:02 PM on Tue Mar 13,2007


                                                        Cache Slots
                     Min  Tgt  Max  WP   Time   Device  ------------------  Used
Partition Name       (%)  (%)  (%)  (%)  (s)    Count      WP       Used    (%)
-----------------   ---  ---  ---  ---  -----  ------  --------  --------  ----
DEFAULT_PARTITION     0   65  100   80    300    1024      2156    395166    95
Lab2                 35   35   35   50    300      56       234      6898    65
```

SRDF/A output

To have the SRDF/A output based on cache partition values, instead of system-wide settings, set `SYMAPI_CP_ANALYSIS_MODE` in the options file to `ENABLE` when running in `analyze` mode.

The cache partition values display automatically when cache partitions are enabled, but only display in `analyze` mode when this options file setting is enabled. The default value is `DISABLE`.

The SRDF/A information displayed in CLI commands will show the percent of cache in use, calculated by the target cache percentage and the write pending limit for the partition of the associated SRDF group.

```
RDFA Information:
            {
            Session Number                     : 39
            Cycle Number                       : 0
            Number of Devices in the Session   : 16
            Session Status                     : Inactive
. . .
            Tracks not Committed to the R2 Side: 0
            Time that R2 is behind R1          : 00:00:00
            R1 Side Percent Cache In Use       :  0
            R2 Side Percent Cache In Use       :  0
. . .
```

Preserving empty partitions

By default, empty partitions or partitions without assigned devices are removed after two hours with no activity. This feature appears in the partition settings output (see the DCP settings output).

To change this feature, use the following form:

symqos -cp -sid *SymmID* set empty preserve

Listing cache partitions

When listing cache partitions, the information can be for a range of devices, a device group, an SRDF group, an array, or a cache partition name. List cache partition information using the following form:

```
symqos -cp [-name PartitionName]

     list [-devs SymDevStart:SymDevEnd [-all]| -rdfg GrpNum]
     list -g DgName [-std] [-bcv] [-vdev]
     list [-sid SymmID] -settings [-usage] [-v]
```

The `-settings` option lists the general cache partition configuration for the array, as shown in the following example:

```
symqos list -pst -sid 237 -settings -v

Symmetrix ID: 000190300237

    Cache Partitioning         : Enabled
    Number of Partitions       : 3
    Max Num of Partitions      : 8
    Min Allowed Target %       : 10
    Max Allowed Target %       : 90
    XRC Partition State        : Disabled
    XRC Partition Name         : N/A
    Empty Partition State      : Remove
    Time of Last Modification  : 10:11:23 AM on Fri Jan 18,2013


    Partition Name             : DEFAULT_PARTITION
    Minimum Cache              : 0%
    Target Cache               : 40%
    Maximum Cache              : 100%
    Write Pending Limit        : 80%
    Donation Time              : 300 seconds
    Devices                    : 00010:0040F
                                 00412:00C14
                                 00D02:00D04
                                 0115C:0116F
                                 01171:0117F
                                 01181:01192
                                 01199:012B2
                                 012B5:012B6
                                 012BD:0133D

    Partition Name             : mktng_cp1
    Minimum Cache              : 5%
    Target Cache               : 10%
    Maximum Cache              : 15%
    Write Pending Limit        : 40%
    Donation Time              : 5 seconds
    Devices                    : 00410:00411
                                 01170
                                 01180
                                 01193:01198
                                 012B3:012B4
                                 012B7:012BC
```

The `-usage` option includes cache usage information. For complete syntax of the `symqos` command, refer to the *EMC Solutions Enabler SYMCLI Command Reference*.

CHAPTER 13
Virtual LUN Technology (Enginuity 5874 through 5876)

This chapter explains how to change device protection or improve array performance using the `symmigrate` CLI command.

- Overview 400
- Migration requirements by features 400
- Creating virtual LUN migration sessions 403
- Migrating to configured space 405
- Migrating to unconfigured space 407
- Migrating to a thin pool 409
- Managing virtual LUN migration sessions 410

Overview

Virtual LUN technology allows storage administrators to quickly move an application's data to other storage within the same array. A migration can change the protection level and disk type of FBA, CKD, and IBM i 512-byte D910 devices. A migration can use the storage of existing devices that do not contain live data or unconfigured disk space.

Virtual LUN migration can be performed from source devices to equally-sized target devices, or disks with free space equal to the source devices. A migration will not disrupt hosts or internal applications. You can use the SYMCLI `symmigrate` command to perform virtual LUN migrations.

A virtual LUN migration may be initiated in the following circumstances:

- Traditional data management involves moving data to a lower tier as it ages and becomes less critical to the business. Protection might change as the data ages.
- Emerging tiered storage configurations include moving data continuously during its lifecycle, either to optimize price/performance or respond to cyclical business changes.

A virtual LUN migration can be performed online or offline.

IMPORTANT

For arrays running HYPERMAX OS 5977, the next-generation FAST technology provides automatic relocation of data through the assignment of Storage Resource Pools. Therefore, Virtual LUN migration technology is not supported on these arrays and migration operations are blocked.

Migration requirements by features

Table 25 lists the hardware, software, disk, and devices requirements for migrating devices using the virtual LUN technology.

Table 25 Virtual LUN requirements by features (page 1 of 2)

Enhanced Virtual LUN Technology	Virtual LUN thin-to-thin mobility	Pool-to-pool migrations	VP (Virtual Pool) compression
VMAX Family arrays running Enginuity 5874, 5875 and 5876	VMAX Family arrays running Enginuity 5875 and 5876	VMAX Family arrays running Enginuity 5876	VMAX Family arrays running Enginuity 5876 Q42012 SR
Solutions Enabler V7.0 and higher	Solutions Enabler V7.2 and higher	Solutions Enabler V7.4 and higher	Solutions Enabler V7.5 and higher
Supported disk types: • Enterprise flash • Fibre Channel • SATA	Supported disk types: • Enterprise flash • Fibre Channel • SATA	Supported disk types: • Enterprise flash • Fibre Channel • SATA	Supported disk types: • Enterprise flash • Fibre Channel • SATA

Table 25 Virtual LUN requirements by features (page 2 of 2)

Enhanced Virtual LUN Technology	Virtual LUN thin-to-thin mobility	Pool-to-pool migrations	VP (Virtual Pool) compression
Supported device types: • Standard VMAX devices, unprotected VMAX devices • Metadevices • FBA, CKD, IBM i 512-byte D910 devices	Supported device types: • Standard VMAX devices, unprotected VMAX devices • Thin devices • Metadevices • FBA, CKD, IBM i 512-byte D910 devices	Supported device types: • Thin devices	Supported device types: • Thin devices
Unsupported device types: • VDEVs (TimeFinder Snap) • DATA devices • Thin devices • Vault devices/SFS/VCM • SAVE devices	Unsupported device types: • VDEVs (TimeFinder Snap) • DATA device • Vault devices/SFS/VCM • SAVE devices		Unsupported device types: Encapsulated devices

Virtual LUN technology restrictions

- If there are CKD thin devices as well as FBA or IBM iseries (AS/400 D910_099) thin devices specified in the same session, use the optional `-force` flag. The CLI responds as follows:
  - If the target pool contains FBA devices, only the FBA and IBM i thin devices will migrate to the target pool.
  - If the target pool contains CKD devices, only the CKD thin devices will migrate to the target pool.
- You cannot migrate protected devices to an unprotected protection type. Migrations of unprotected source devices to unprotected target are supported.
- You cannot migrate to the same protection type (RAID 5 (3+1) to RAID 5 (3+1)) and the same disk group number (disk group 1 to disk group 1). The desired protection type or disk group number must be different from what exists on the source devices. This restriction is for disk group provisioned devices.
- The target devices cannot be in any state where the possibility that data is currently being replicated, for example RDF, clone, or snap devices. These devices are blocked from migration. This restriction is for disk group provisioned devices.
- You cannot migrate a thin device to a standard device, or a standard device to a thin device.
- Thin devices must be in the `Bound` state to be migrated. If they are in any other state, an error returns. If the thin device is not already bound to the target pool, it will be bound to that pool after the migration completes.
- Thin devices that are currently involved in an active migration session are not eligible to transition to other thin device states until the migration completes and the session is terminated.

Virtual LUN VP mobility

Virtual LUN VP mobility allows thin devices to be moved between pools. You can request that, for a given list of devices, the data tracks be moved from their current pool to a target pool. In addition, if the devices themselves are not already bound to the target pool, they will be rebound to the target pool.

To implement this thin-to-thin mobility, the `symmigrate` command provides a target type, `-tgt_pool`, as the destination for a migration. The destination pool is specified with `-pool` *`PoolName`*.

To run a thin migration, all input source devices must be thin devices. Thin migrations will support up to 20,000 devices per migration.

Prior to migrating any allocations, the total allocations for all relevant thin devices are summed, and must be less than the remaining free tracks in the target pool or an error returns and the migration does not take place.

Device pool-level migrations

Migrating allocations from one pool to another pool can be specified for a set of thin devices. A pool-level migration of thin devices does not change the binding of that thin device to the target pool, therefore the rebind option must be done manually, if the rebind is desired.

This functionality differs from the full migration of a TDEV in that not all the TDEV allocations will be migrated, just the allocations that are in the source pool. Any allocations in pools other than the source pool will not be migrated to the target in this pool-to-pool migration.

VP compression guidelines for migration

The `symmigrate` command allows the migration of thin devices to and from pools that have compression enabled. The following describes thin device migration behavior between source and target thin pools, based on VP compression status of the pool:

- If VP compression is enabled for both the source and target pool, the compressed tracks migrate to the target as compressed tracks. The target pool's utilized space increases by the compressed size and the free space decreases by the compressed size. The source pool's utilized space decreases by the compressed size and the free space increases by the compressed size.
- If VP compression is enabled for the source pool, but disabled for the target pool, compressed devices must be uncompressed prior to migration. Solutions Enabler does not allow a compressed device to be migrated to a pool where compression is disabled.
- If VP compression is disabled for both the source and target pool, the uncompressed (allocated) tracks migrate to the target pool as uncompressed tracks (allocated). The target pool's utilized space increases by the uncompressed (allocated) size and the free space decreases by the uncompressed (allocated) size. The source pool's utilized space decreases by the uncompressed (allocated) size and the free space increases by the uncompressed (allocated) size.

FTS support

The `symmigrate` CLI supports both encapsulated and externally provisioned devices. Encapsulated devices cannot be used as targets of a VLUN migration and cannot be migrated using the configured space option. Migrations using device pairs for external provisioned devices is fully supported.

Creating virtual LUN migration sessions

The `symmigrate` CLI provides the actions listed in Table 26 to create and manage migration sessions.

Table 26 Action descriptions for symmigrate command

symmigrate action	Description
`establish`	Starts a new session and begins the synchronizing process.
`list`	Lists all the sessions for a given array, or for all arrays.
`query`	Queries for the status of the specified session.
`terminate`	Terminates a session and removes it from the array.
`validate`	Verifies that the information provided is currently allowed. No changes will be made to the device(s) and the command can optionally output a file containing device pairs (when using configured space).
`verify`	Verifies that a session is in a specified state.

Note: During the migration the source device will be available for all local and remote replication. The target device will be set by the system as user not ready for the duration of the migration and all operations to set the device to ready will be blocked.

Specifying source devices

You can specify source devices by using the following:

- A device group
- A storage group
- A source file (containing a list of source devices or device pairs). Refer to "Using files" on page 404.

Specifying target criteria

Target criteria can be specified as configured space or unconfigured space. A target disk group is specified, as well as the target protection type: Unprotected, RAID1, RAID 5 (3+1 or 7+1), or RAID 6 (6+2 or14+2), unless using pairs in a file. The target protection type and disk group is assumed when using a device file.

The `symmigrate` command now supports a new target type, `-tgt_pool`, as the destination for a thin migration. The destination pool is specified with `-pool` *`PoolName`*.

Using files

Device file for source and target pairing

A device file (*DevFile*) is a text file that contains pairs of source and target devices and one pair per line in the file.

Note: Make sure all the devices in the device file match the type of migration specified.

For example:

```
10     20
12     22
14     27
```

For example, a device to a metadevice:

```
15    28, 29, 30, 31,\
   32, 33, 34, 39
```

A metadevice (source) to metadevice (target) file is also allowed, if the device sizes are equal, and all members are also equal in size and order.

The line continuation character of '\' can be used to wrap lines.

Source file

A source file (*SrcFile*) is a text file that contains only source devices and one device per line in the file.

For example:

```
9A
9E
9F
```

Establishing a migration session

The `symmigrate establish` command creates the migration session and begins the migration of the source devices to the new target location. You must provide the target protection type and disk group number, unless using a device pair file or thin devices. This is the point in the migration process that the mirrors are moved to the source device from the target, if using configured space, or are created if using unconfigured space. The migration of thin devices is similar to the migration of regular devices. You can specify device groups, storage groups, or devices ranges as source thin devices. To run a thin migration, all input source devices must be thin devices, unless the source is a device group or storage group. Thin migrations will support up to 20,000 devices per migration.

Like all other CLI interfaces, the migration control operation will operate on the metahead only. Only the metahead is needed for the control operation; the meta members will be determined by the system. In a configured disk space migration, if the source device is a metadevice, non metadevices can be chosen to match the source device provided that the devices are matched in size and quantity to the meta members. In this case, the output from the `validate` command will be a device-devices list instead of a device-device pair.

The source device will be the meta head and the target devices will be a comma separated device list. Alternatively, if there is an available metadevice to use as a target, it can be specified in the standard pair file format.

Note: CKD striped metadevices can only migrate to unconfigured space, and the target protection type is restricted to RAID-1.

Validating a migration session

The `symmigrate validate` command is an optional command, that can be used in two different ways. The command can be used as a simple positive/negative test to determine if a migration will succeed. This will be denoted by the return value of the command. The command can also be used when migrating to configured space to output a file that contains the exact pairs of devices that will be used for the migration. This output can be used to verify that the command will produce the exact migration that is desired by the user and to modify the resulting file if it is not.

Any `symmigrate` operations performed on a parent storage group will be performed on all devices contained by all child storage groups of a specified parent storage group. Those operations include:

- `validate`
- `establish`
- `terminate`

Note: The file output is only a listing of what the system chose to do at the time the `validate` command was run. To guarantee that the exact pairs that were returned are used, run the `symmigrate establish` command with the file that was output by the `validate` command.

Migrating to configured space

A migration to configured space can be performed using a device file or by specifying target criteria.

The following sections explain how to perform each type of migration operation.

Migrating using a device file

Use the following syntax create a migration session using a device file:

```
symmigrate [-v][-i Interval] [-c Count] [-noprompt]
         -name SessionName
         -file DevFile -sid SymmID
  validate
  establish

The symmigrate command would be as follows:
   symmigrate validate -sid 123 -f input.txt -name DevMigration
   symmigrate establish -sid 123 -f input.txt -name DevMigration
```

Migrating using target criteria

To perform a migration to configured space, follow this form:

```
symmigrate [-v]
           [-i Interval] [-c Count] [-noprompt]
           -name SessionName
           <-file SrcFile -sid SymmID |
            -sg SgName -sid SymmID |
            -g DgName [-bcv | -tgt]>
           -tgt_config -tgt_dsk_grp DskGrp | name:DskGrpName
           <-tgt_unprotected                 |
            -tgt_raid1                       |
            -tgt_raid5 -tgt_prot <3+1 | 7+1> |
            -tgt_raid6 -tgt_prot <6+2 | 14+2>>

      validate [-outfile OutputFile]
      establish
```

Where:

`-name` — Indicates a session name when validating or establishing a session.

The source devices can be identified by one of the following:

- A file, *SrcFile*, which requires the array ID
- A storage group (`-sg`), which requires the array ID
- A device group, with an optional pointer to the BCV or TGT list

The target criteria is specified as follows:

`-tgt_config` — Tells the array to look for configured space.

`-tgt_dsk_grp` — Identifies the disk group number or name for the migration. The source devices will be in that disk group when the migration is complete.

Protection — The following options specify the protection type that the devices will have when the migration is complete:

`-tgt_unprotected` — Unprotected. This option is only valid if the source devices are unprotected.

`-tgt_raid1` — RAID 1 protection.

`-tgt_raid5` — RAID 5 protection with either `-tgt_prot 3+1` or `7+1`.

`-tgt_raid6` — RAID 6 protection, with either `-tgt_prot 6+2` or `14+2`.

Error reporting

The `establish` and `validate` migration operations examine all target devices supplied in the user input and return a message for each, indicating whether the device was selected or skipped. The message lists the device number and if skipped, the reason why it was skipped. A device may be skipped when the device is (or has):

- An invalid configuration
- A local or remote replication or migration session
- Host visible
- Capacity, emulation, protection type, or block size mismatch

- Reserved

The `symmigrate validate` command can be used to produce a file that will indicate which devices the system identifies as the target *at the time* the `symmigrate validate` command is run.

For the following example, the device input file named `test1` contained:

```
00A20 00C90
00A21 00C91
00A22 00C92
00A23 00C93
00A24 00C95
00A25 00C99
```

The following output shows the new messages from a failed `symmigrate establish` operation for the configured devices shown above in the file `test1`.

The -v (verbose) option that triggers the expanded output with establish and validate operations:

```
symmigrate -file test1 -sid 123 -name test1 establish -nop -v

`Establish' operation execution is in progress for
the device list in device file `test1'. Please wait...

STARTING a Migrate `ESTABLISH' operation.

Source devices:
00A20:00A25 [SELECTED]

Target devices:
00C90:00C93 [SKIPPED - Local or remote replication or migration]
00C95       [SELECTED]
00C99       [SKIPPED - Host visible]
The Migrate `ESTABLISH' operation FAILED.

There are not enough target devices to complete the specified migration
```

Migrating to unconfigured space

This section explains how to perform a virtual LUN migration to unconfigured space.

Use the following form when migrating devices to unconfigured space:

```
symmigrate [-v]
         [-i Interval] [-c Count] [-noprompt]
         -name SessionName
         <-file SrcFile -sid SymmID |
          -sg SgName -sid SymmID |
          -g DgName [-bcv | -tgt]>

         <-tgt_unconfig -tgt_dsk_grp DskGrp | name:DskGrpName

         <-tgt_unprotected                 |
          -tgt_raid1                       |
          -tgt_raid5 -tgt_prot <3+1 | 7+1> |
          -tgt_raid6 -tgt_prot <6+2 | 14+2>>
validate
establish
```

Where:

`-name` — Indicates a session name when validating or establishing a session.

The source devices can be identified by one of the following:

- A file, *SrcFile*, which requires the array ID
- A storage group (`-sg`), which requires the array ID
- A device group, with an optional pointer to the BCV or TGT list

The target criteria is specified as follows:

`-tgt_unconfig` — tells the array to look for unconfigured space.

`-tgt_dsk_grp` — identifies the disk group number or name for the migration. The source devices will be in that disk group when the migration is complete.

Protection — the following options specify the protection type that the devices will have when the migration is complete:

`-tgt_unprotected` — Unprotected. This option is only valid if the source devices are unprotected.

`-tgt_raid1` — RAID 1 protection.

`-tgt_raid5` — RAID 5 protection with either `-tgt_prot 3+1` or `7+1`.

`-tgt_raid6` — RAID 6 protection, with either `-tgt_prot 6+2` or `14+2`.

An example of a virtual LUN migration of devices in a device group `-tgt` list to unconfigured space, with RAID 5 3+1 protection follows:

```
symmigrate validate
      -g mydg -tgt
      -tgt_raid5 -tgt_prot 3+1 -tgt_dsk_grp 5
      -tgt_unconfig -name DGunconfig

 symmigrate establish
      -g mydg -tgt
      -tgt_raid5 -tgt_prot 3+1 -tgt_dsk_grp 5
      -tgt_unconfig -name DGunconfig -nop
```

Error reporting

The `establish` and `validate` migration operations examine all target devices supplied in the user input and return a message stating whether the device was selected or skipped. The message lists the device number and if skipped, the reason why it was skipped. A device may be skipped when the device is (or has):

- An invalid configuration
- A local or remote replication or migration session
- Host visible
- Capacity, emulation, protection type, or block size mismatch
- Reserved

For the following example, the device input file contained:

```
0A33
```

The following output shows the message from a successful `symmigrate establish` for the unconfigured devices.

The -v (verbose) option that triggers the expanded output with establish and validate operations:

```
symmigrate -file test1 -sid 123 -name test1 establish -tgt_unconfig   -tgt_raid1
   -tgt_dsk_grp 1 -nop -v

`Establish' operation execution is in progress for
the device list in device file `test1'. Please wait...

STARTING a Migrate `ESTABLISH' operation.

Source devices:
00A33      [SELECTED]

    Establish Migration.............................................Started.
    Establish Migration.............................................Done.

The Migrate `ESTABLISH' operation SUCCEEDED.
```

Migrating to a thin pool

The Virtual LUN VP mobility feature allows thin devices to be moved between pools. The source thin devices can be specified in a file, device group, or storage group. The target location is specified using a new target type, `-tgt_pool`. The source pool should be removed from FAST control before moving allocations to the target pool.

Use the following form when migrating devices to a thin pool:

```
symmigrate [-v] [-i Interval] [-c Count] [-noprompt] -name SessionName
         <-file SrcFile -sid SymmID |
          -sg SgName -sid SymmID |
          -g DgName [-bcv | -tgt]>
          -src_pool PoolName
          -tgt_pool -pool PoolName
   establish
   validate
```

Where:

-`src_pool` — Specifies the name of the pool to be used as the source for a thin migration to a target thin pool. Only allocations within the source pool for the designated set of thin devices will migrate to the target pool.

-`tgt_pool` — Indicates this is a thin migration. This option must also include the thin pool name (`-pool` *`PoolName`*).

The other options and parameters are identical to those defined in "Migrating to configured space" on page 405 and "Migrating to unconfigured space" on page 407, including the verbose error reporting (`-v`).

Viewing a thin migration

The output display for a thin migration includes the migration type (`Flags`), as shown in the following example:

```
symmigrate query -name sample_test -sid 266

Symmetrix ID: 000194900266
```

```
                Invalid    Status     Done Flags
Src   Tgt       Tracks   SRC => TGT   (%)    T   Session Name
----- ----- -------- ------------ ---- ----- ----------------
00437   N/A        0 Migrated      100   P   sample_test

Total         --------
 Tracks             0
 MB(s)            0.0
 Done(%)          100

Flags:
(T)ype: V = The session is a Virtual Provisioning migration session.
        C = The session is a configured migration session.
        U = The session is an unconfigured migration session.
        P = The session is a Virtual Provisioning pool migration session.
```

The `-detail` option provides both the flag and target description to the output display. The `TGT Description` field contains the target pool name if the session is a thin migration or the target protection type if the session is a non-thin migration.

The following is an example of `symmigrate query` with the `-detail` option:

```
symmigrate query -name sample_test -sid 266 -detail

Symmetrix ID: 000194900266

            Invalid    Status    Done Flags                Disk
Src   Tgt     Tracks  SRC => TGT   (%)   T      SRC Pool   Grp          TGT Description         Session Name
----- ----- -------- ------------ ---- -----  ------------ ---- ---------------------------- -------------------
00437   N/A        0 Migrated      100   P    pool1        01   pool2                           sample_test

Total     --------
 Tracks          0
 MB(s)         0.0
 Done(%)       100

Flags:
(T)ype: V = The session is a Virtual Provisioning migration session.
        C = The session is a configured migration session.
        U = The session is an unconfigured migration session.
        P = The session is a Virtual Provisioning pool migration session.
```

Managing virtual LUN migration sessions

The `symmigrate` command allows you to `list`, `verify`, `query`, and `terminate` the migration sessions by using the session name, or by using the session name and specifying the filename, device group, or storage group.

Note: If a session has existed for two weeks without being listed or queried, it will automatically be removed, at which point it will not be returned by a `query` or `list` command.

The array device lock will be held on the source device(s) only when the `establish` and `terminate` operations are in progress. The lock will be released from the time the migration enters the Sync In Progress state until you terminate the session. When the session is terminated, the lock will be taken again.

A limit of 16 concurrent migration and configuration change sessions can be run for devices in disk group migrations. For thin devices in thin pool migrations, the limit is 128.

Listing sessions

Use the `symmigrate list` command to display all the existing migration sessions for a specified array, or all arrays, as follows:

```
symmigrate [-v] [-i Interval] [-c Count]
list [-sid SymmID] [-names] [-detail]
```

For example, to display the migration sessions currently on array 266, enter:

`symmigrate -sid 266 list`

The following is an example of the output display:

```
Symmetrix ID: 000194900266

            Invalid   Status      Done Flags
Src   Tgt   Tracks   SRC => TGT   (%)    T   Session Name
----- ----- -------- ------------ ---- ----- ----------------
00200 N/A          0 Migrated      100   V   testb200
00202 N/A          0 Migrated      100   V   testb202
  . . .
0027E N/A          0 Migrated      100   V   testb27e
00437 004F8        0 Migrated      100   V   sample_test

Total       --------
 Tracks            0
 MB(s)           0.0
 Done(%)         100

Flags:
(T)ype: V = The session is a Virtual Provisioning migration session.
        C = The session is a configured migration session.
        U = The session is an unconfigured migration session.
        P = The session is a Virtual Provisioning pool migration session.
```

The `-detail` option includes the source and target information for each session in the display, as shown in the following sample output:

`symmigrate -sid 266 list -detail`

```
Symmetrix ID: 000194900266

           Invalid    Status    Done Flags              Disk
Src  Tgt    Tracks  SRC => TGT   (%)   T     SRC Pool   Grp          TGT Description          Session Name
---- ---- -------- ------------ ---- ----- ------------ ---- ------------------------------ ----------------
0200  N/A        0 Migrated      100   V   N/A          01   testb2                         testb200
0202  N/A        0 Migrated      100   V                01   testb2                         testb202
 . . .
027E  N/A        0 Migrated      100   V                01   testb2                         testb27e
0437  04F8       0 Migrated      100   V                01   testb1                         sample_test

Total     --------
 Tracks          0
 MB(s)         0.0
 Done(%)       100

Flags:
(T)ype: V = The session is a Virtual Provisioning migration session.
        C = The session is a configured migration session.
        U = The session is an unconfigured migration session.
        P = The session is a Virtual Provisioning pool migration session.
```

Querying sessions

Use the `symmigrate query` command to display the status of a specific migration session, or all migration sessions, as follows:

```
symmigrate [-v] [-i Interval] [-c Count]
            -name SessionName -sid SymmID
```

```
query [-detail][-summary]
```

The `query` command can be used with the session name only, with the addition of the array ID.

Query with -summary option

When the `-summary` option is specified, a summary listing of session states and the number of sessions in each state will be included in the output from each command in order to assist with interpreting the command output.

Used with `symmigrate query`, the `-summary` option replaces the table(s) listing all sessions and their current session states with an abbreviated listing that shows all possible session states and the number of sessions within the scope of the query that are in each state.

The `Total` summary that shows total invalid tracks will be formatted differently when `-summary` is specified.

The estimated time to completion will be shown when:

- At least one of the `-i` and/or `-c` options is specified.
- There has been a change in the invalid track counts since the previous iteration.

The following shows the output from `symmigrate query -summary`:

```
symmigrate query -name sess -summary -i 5  c 2

Session name: sess

  Migration Session State       Count
  -----------------------      ------
  CreateInProg                      0
  SyncInProg                        2
  Synchronized                      0
  MigrateInProg                     0
  Migrated                          0
  Failed                            0
  Invalid                           0
  -----------------------      ------
  Total                             2

                                 Track(s)    MB(s)
                               -----------  -------
  Total Invalid                    26863   1678.9


Session name: sess

  Migration Session State       Count
  -----------------------      ------
  CreateInProg                      0
  SyncInProg                        1
  Synchronized                      1
  MigrateInProg                     0
  Migrated                          0
  Failed                            0
  Invalid                           0
  -----------------------      ------
  Total                             2

                                 Track(s)    MB(s)
                               -----------  -------
```

```
Total Invalid                        11153    697.1

Synchronization Rate         :     109.1 MB/S
Estimated time to completion :  00:00:06
```

Synchronization rate and estimated time to completion are shown only when `-i` and/or `-c` is specified and the number of invalid tracks has changed since the previous iteration.

Verifying sessions

The `symmigrate verify` command allows you to determine if a migration session is in a specific state. The syntax follows:

```
symmigrate -sid SymmID -name SessionName [-i Interval] [-c Count]

   verify [-createinprog | -syncinprog | -synchronized
           -migrateinprog | -migrated | -failed | -invalid]
          [-summary]
```

A virtual LUN migration session is complete when it is in the Migrated state.

For example, to verify whether a migration session named `migrate1` is in the migrated state, enter:

```
symmigrate -sid 258 -name migrate1 verify -migrated -i 30
```

The output is similar to the following:

```
NONE of the devices are in the 'Migrated' state.
NONE of the devices are in the 'Migrated' state.
……................................................................................
NONE of the devices are in the 'Migrated' state.
ALL of the devices are in the 'Migrated' state.
```

Verifying sessions with the -summary option

Used with `symmigrate verify`, the `-summary` option provides the same information as provided by the `symmigrate query -summary` command. That identical information precedes the single line summary provided by `symmigrate verify`.

Terminating sessions

A migration session remains active until it is terminated. When the status of the devices in the session is Migrated, the session is complete and can be terminated. Thin migrations can be terminated prior to completion.

To terminate a session, use the following form:

```
symmigrate -name SessionName -sid SymmID>

   terminate [-noprompt]
```

Viewing the audit log

For any control command, the API will log the start and end of the operation and any errors that may occur. Table 27 lists the information that is saved in the audit log based on the target selection type.

Table 27 Audit log session information

Target selection type	Information in the audit log
Device file	Session name, source devices, target disk group, target protection type, and target devices
Configured/unconfigured space with no specified devices	Session name, source devices, target disk group, and target protection type
A thin pool	Session name, source devices, target protection type, and target pool

For information about the `symaudit` command, refer to the *EMC Solutions Enabler SYMCLI Command Reference*.

PART 2
Querying and Reporting

Part 2 of this guide describes the SYMCLI query commands and various tools for monitoring and managing storage array performance.

Chapter 14, "Configuration Query Operations," describes how to use the SYMCLI to collect and display configuration data for array devices, disks, and virtual environments.

Chapter 15, "Statistics," describes how to use SYMCLI to retrieve performance-related statistics such as the number of I/O requests and throughput activities for array devices.

Chapter 16, "Change Tracker (Enginuity 5876 or lower)," explains how to monitor storage array data objects for change within your storage environment.

Chapter 18, "Managing Time Windows (Enginuity 5876 or lower)," explains how to add, remove, convert, and display time windows.

Chapter 17, "Optimizing Array Performance (Enginuity 5876 or lower)," describes how to improve array performance using SYMCLI Optimizer commands.

Chapter 19, "Performing Double Checksum Operations (Enginuity 5876 or lower)," describes how to minimize the impact of I/O errors on database consistency during I/O transfers using the Double Checksum SYMCLI commands.

Chapter 20, "Managing Network IPsec (Enginuity 5876 or lower)," describes the IPsec standard and how to manage IPsec network policies using the SYMCLI IPsec component.

Chapter 21, "Events and Logs," describes how to configure, manage, and query storage array events and logs.

Chapter 22, "XML Structured Output," describes the SYMCLI XML output option.

CHAPTER 14
Configuration Query Operations

This chapter describes how to use the SYMCLI to collect and display configuration data for DMX and VMAX Family array devices, disks, and virtual environments.

- Configuration data 418
- SCSI-level data 418
- Array-level data 422
- Device-level data 449
- Disk-level data 475

Configuration data

The Solutions Enabler SYMCLI provides various commands that are used to query different levels of the storage environment. These levels include:

- SCSI-level — Data can be returned at the SCSI level, whereby the SYMCLI issues SCSI INQUIRY and SCSI READ CAPACITY to return low-level physical device data (such as vendor, configuration, and basic configuration) and host HBA information (such as vendor, model, firmware, and basic configuration).
- Array-level — Array queries allow you to return further detailed data about the configuration of one or all arrays. The data returned includes host relationship information (local or remote), cache size, and number of devices. Further details include vendor specific configuration information, ports, flags, adapters, Data at Rest Encryption data, pool information, and environment data.
- Device-level — As you go deeper into your storage array, you can gather more detailed information about the devices in a given array (or all arrays, either locally or remotely attached). Device-level data includes capacity, cache, emulation, configuration, group, and usage information that is critical to all other storage management operations.
- Disk-level — Returns detailed information about the raw disks from which your storage devices are derived, including detailed identification, vendor, and capacity.
- Virtual environment — Returns configuration information about a VMware or Microsoft Hyper-V environment, such as the type and capacity of their storage pools.

SCSI-level data

The `syminq` command is used to obtain SCSI disk device information using a SCSI INQUIRY command, and optionally SCSI READ CAPACITY command, on host HBAs and one or all locally attached physical devices.

The scope of an inquiry to EMC arrays, StorageWorks, or HDS devices can be controlled. When specifying an array, you can issue the `syminq` command to control what data is returned and its presentation. The *EMC Solutions Enabler SYMCLI Command Reference Guide* describes the various options available with the command.

You can use the `syminq` command to list all of the HBAs on the local host. Using the options available, the scope of this request can be limited to list only the Fibre HBAs, SCSI HBAs, iSCSI HBAs, or HBAs derived from a SNIA API query. In the UNIX environment, if the SNIA libraries are unavailable, HBA information is obtained from the lost log files.

You can limit the returned results to devices with target mapping information (only devices mapped through fibre HBAs using the `-mapinfo` option), or to devices given device identifiers by a user or application by using the `-identifier` option. Identifiers are limited to `device_name`, `nice_name`, `hp_id`, and `vms_id`.

Note: Data returned from issuing any `syminq` command is not stored in the configuration database.

SCSI device information

The following is an example of a SCSI inquiry of all devices listed by physical device name:

syminq

```
      Device              Product                    Device
---------------- -------------------------- ---------------------fsy
Name      Type   Vendor    ID           Rev  Ser Num      Cap (KB)
---------------- -------------------------- ---------------------
/dev/sda         MAXTOR    ATLAS10K4_3* DFL0 B2CS5FSM     35916548
/dev/sdb  GK     EMC       SYMMETRIX    5772 150009E020       5760
/dev/sdi  GK     EMC       SYMMETRIX    5772 15000A7020       5760
/dev/sdj  GK     EMC       SYMMETRIX    5772 15000A8020       5760
/dev/sdk  GK     EMC       SYMMETRIX    5772 15000A9020       5760
/dev/sdl  GK     EMC       SYMMETRIX    5772 15000AA020       5760
/dev/sdm  GK     EMC       SYMMETRIX    5772 15000AB020       5760
/dev/sdn  GK     EMC       SYMMETRIX    5874 0700022000       5760
/dev/sdo  GK     EMC       SYMMETRIX    5874 0700023000       5760
/dev/sdp  GK     EMC       SYMMETRIX    5874 0700024000       5760
/dev/sdq  GK     EMC       SYMMETRIX    5874 0700025000       5760
. . .
```

Devices listed by array ID

The following is an example of a SCSI inquiry of array devices listed by ID:

syminq -symmids

```
      Device          Symmetrix               Device
---------------- --------------------- ---------------------
Name      Type   ID                Rev  Ser Num      Cap (KB)
---------------- --------------------- ---------------------
/dev/sdb  GK     000190300215     5772 150009E020       5760
/dev/sdc  GK     000190300215     5772 150009F020       5760
/dev/sdd  GK     000190300215     5772 15000A2020       5760
/dev/sde  GK     000190300215     5772 15000A3020       5760
/dev/sdf  GK     000190300215     5772 15000A4020       5760
/dev/sdg  GK     000190300215     5772 15000A5020       5760
/dev/sdh  GK     000190300215     5772 15000A6020       5760
. . .
```

Devices without capacity

The following is an example of a SCSI inquiry of arrays without issuing a SCSI READ CAPACITY:

syminq -symmids -nocap

```
Device           Symmetrix             Device
---------------- --------------------- ----------
Name      Type   ID                Rev  Ser Num
---------------- --------------------- ----------
/dev/sdv  BCV    000190300516     5771 1600028000
/dev/sdw  BCV    000190300516     5771 1600029000
/dev/sdx  BCV    000190300516     5771 160002A000
/dev/sdy  BCV    000190300516     5771 160002B000
/dev/sdz  BCV    000190300516     5771 160002C000
```

Devices with WWN

The following is an example of a SCSI inquiry of arrays returning the WWN for each device:

```
symiq -wwn

         Device                                          Device
----------------------------- ---------------- --------------------------------
Name                     Num   Array ID         WWN
----------------------------- ---------------- --------------------------------
/dev/sda                   N/A N/A              N/A
/dev/sdb                 0005E 000194900306     60000970000194900306533030303545
/dev/sdy                 000E6 000194900306     60000970000194900306533030304536
```

The following is an example of how the `-colon` option can be used to return a list of devices with WWN for each device using colons as spacers:

```
symiq -wwn -colons

         Device                                          Device
----------------------------- ----------------
-----------------------------------------------
Name                     Num   Array ID         WWN
----------------------------- ----------------
-----------------------------------------------
/dev/sda                   N/A N/A              N/A
/dev/sdb                 0005E 000194900306     60:00:09:70:00:01:94:90:03:06:53:30:30:30:35:45
/dev/sdy                 000E6 000194900306     60:00:09:70:00:01:94:90:03:06:53:30:30:30:45:36
```

Pdevfile format

The following is an example of a SCSI inquiry of arrays returning a list of device names in a format suitable for use as `pdevfile`:

```
symiq -pdevfile

# Symm_id    pdev               dev dir  dir_port
000187900771 /dev/rdsk/c2t0d0s2 0017 15D  0
000187900771 /dev/rdsk/c2t0d1s2 0018 15D  0
000187900771 /dev/rdsk/c2t0d2s2 0019 15D  0
000187900771 /dev/rdsk/c2t0d3s2 001A 15D  0
```

HBA information

To obtain a list of all of the local host's HBAs, use the `symiq hba` command entry. The returned data can be filtered by using one or a combination of available options, such as to list only the Fibre (`-fibre`), SCSI (`-scsi`), or iSCSI (`-iscsi`) HBAs. In addition, the `-snia` option allows you to return HBA information using the native SNIA SMI-S Provider rather than by issuing a SCSI inquiry. The returned data will be exactly the same as the data returned with a `-fibre` SCSI inquiry; the only difference is how the data was obtained.

For example, to obtain a list of the local host's HBAs, enter:

```
symiq hba

  Host Name : api171

  HBA Type            : FibreChannel
  HBA Name            : Emulex-LPe11002-E-1
```

```
Vendor              : Emulex Corporation
Model               : LPe11002-E
Serial Number       : VM63487963
Firmware Version    : 2.50A4 (Z2F2.50A4)
Driver Version      : 8.1.10.3; HBAAPI(I) v2.1.d, 07-28-06
Node WWN            : 0000000000000000
Number of Ports     : 2

    Port WWN            : 10000000c9594dce
    Port name           : /sys/class/scsi_host/host1
    Port type           : NPort
    Port FCID           : 7434496
    Port speed          : 4gbit
    Supported speed     : 4gbit
    Port state          : Online
    Supported COS       : 00000008
    Supported FC4 types : 00000100000000010000000000000000000000000000000000000000000000
    Active FC4 types    : 00000100000000010000000000000000000000000000000000000000000000
    Max frame size      : 2048

    Port WWN            : 10000000c9594dcf
    Port name           : /sys/class/scsi_host/host2
    Port type           : Unknown
    Port FCID           : 0
    Port speed          : Unknown
    Supported speed     : 4gbit
    Port state          : LinkDown
    Supported COS       : 00000008
    Supported FC4 types : 00000100000000010000000000000000000000000000000000000000000000
    Active FC4 types    : 00000100000000010000000000000000000000000000000000000000000000
    Max frame size      : 2048
```

Note: HBA returned data will vary slightly when the `-scsi` or `-iscsi` option is specified.

Mapping information

The `syminq` command returns target mapping information when the `-mapinfo` option is specified. The following is the `syminq -mapinfo` syntax:

```
syminq -mapinfo[PdevName]
    [-sym[-powerpath]|-hds|-storworks]
    [-cache | -nocache][-colons][-winvol]
```

When the `syminq -mapinfo` command is called, the physical device name, HBA port WWN, and target port WWN is output for each host device. The following is an example of the `syminq -mapinfo` command output:

syminq -mapinfo

```
               Device                          Target Mapping
------------------------------------ ----------------------------------
Name                                 HBA Port WWN     Target Port WWN
------------------------------------ ----------------------------------

/dev/rdsk/c0t0d0s2                   N/A              N/A
/dev/rdsk/c1t1d0s2                   N/A              N/A
/dev/rdsk/c2t5000097208013558d0s2    210000e08b80a234 5000097208013558
/dev/rdsk/c2t5000097208013558d1s2    210000e08b80a234 5000097208013558
/dev/rdsk/c2t5000097208013558d2s2    210000e08b80a234 5000097208013558
/dev/rdsk/c2t5000097208013558d3s2    210000e08b80a234 5000097208013558
/dev/rdsk/c2t5000097208013558d4s2    210000e08b80a234 5000097208013558
```

Device identifiers

To list the array device identifiers assigned to devices by the user or other applications, use the `syminq -identifier` option and specify one of the four options: `device_name`, `nice_name`, `hp_id`, or `vms_id`.

Example: **`syminq -identifier vms_id`**

```
      Device                                  Device
--------------- ------------------------ ----------------
Name        Num Vendor          Array ID VMS ID
--------------- ------------------------ ----------------
/dev/sdb  0005E EMC        000194900306 94
/dev/sdc  0005F EMC        000194900306 95
/dev/sdy  000E6 EMC        000194900306 230
```

Array-level data

A storage array contains a complex set of data. The `symcfg` command provides arguments and options to use to pinpoint specific components and access the information you need. The storage array data includes the following:

- Storage arrays — A list of all arrays attached to the host and configuration details about them.
- Registered applications — Applications registered with SYMAPI that have accessed all or specified arrays to which your host is connected.
- Host connections — Detailed information about the hosts that have accessed a array, the host node name and the array ID, director/port information, and array capacity.
- Director information — Configuration and status information about all directors of a given array including address, port, and status information. Details can be limited to a specific type of director.
- SELs (Symmetrix External Locks) and Semaphores — Status information on SELs (by number or type) and SYMAPI semaphores.
- Cache management — Information on the LRU cache, including the cache slots that each LRU occupies, and the percentage of the total cache utilized.
- Network services — A list of network services available to SYMAPI client applications.
- Environment data — Detailed information on the memory boards, and the array's environmental data, including fans and power supplies, can be obtained.
- CU Images — Information for mainframe users.
- Microcode patches — A list of installed operating environment for array patches.
- Bay location descriptions — Information listing the bay names and location details of the array.
- Data at Rest Encryption (Enginuity 5875 or 5876) — Information about hardware-based, on-array, back-end encryption for arrays.

Listing arrays

To query storage environment configuration data, the arrays in the storage environment must first be identified. Each array has a serial number (SID) that is used to uniquely identify it. Using SYMCLI, you can obtain a list of all the accessible arrays by SID including the model number and the number of accessible devices.

You can use SYMCLI commands to obtain detailed configuration information about a specific array. This high-level information (such as the SID) can be used with subsequent commands to obtain configuration information for each of the directors and devices within an array.

Issue the `symcfg list` command to obtain a list of all array IDs for the arrays connected to your host, with one entry for each array accessible from the host.

```
symcfg list

                          S Y M M E T R I X

                                      Mcode    Cache      Num Phys  Num Symm
 SymmID       Attachment  Model      Version  Size (MB)  Devices   Devices

 000190100097 Local       DMX3-24    5771       98304        20     32088
 000000006206 Remote      2000P-M2   5671       32768         0      4139
 000187900035 Remote      800-M2     5671        8192         0      1314
 000187900041 Remote      800-M2     5671        8192         0       960
 000190300174 Remote      DMX3-6     5771       16384         0       335
 000190300175 Remote      DMX3-6     5771       16384         0       914
 000190300184 Remote      DMX3-6     5771       16384         0       417
```

To obtain a detailed list for a specific array, enter:

```
symcfg list -v -sid 2300

Product Model                        : VMAX40K
         Symmetrix ID                         : 000195602300

    Microcode Version (Number)           : 5876 (16F40000)
    Microcode Registered Build           : 0
    Microcode Date                       : 10.16.2012

    Microcode Patch Date                 : 10.16.2012
    Microcode Patch Level                : 159

    Symmwin Version                      : 30
    Enginuity Build Version              : 5876.159.30
    Service Processor Time Offset        : - 00:00:02

    Cache Size (Mirrored)                :   18432 (MB)
    # of Available Cache Slots            :  253968
    Max # of System Write Pending Slots  :  152777
    Max # of DA Write Pending Slots      :       0
    Max # of Device Write Pending Slots  :    7638
    Max # of Replication Cache Slots     :   43174
    Replication Cache Usage (Percent)    :      10

        . . .
```

Note: For Enginuity releases prior to 5875, the Replication Cache Maximum is reported as N/A. For compatibility modes earlier than 7.6, the Replication Cache Maximum is not reported. Compatibility mode specifies the command output reporting style to be compatible with prior SYMCLI versions. Refer to "Setting compatibility" on page 31.

Listing array bay locations (Enginuity 5876 or lower)

To list the bay names and location information of array 166, enter:

```
symcfg list -bay_location -sid 166

Symmetrix ID               : 000192600166

    Bay Name     : SB-1
    Bay LED state: On
    Bay Location : This is the system bay located in the upper side of the B lab near
    the water cooler. Walk through the double glass doors and take a left turn at the
    air conditioning unit. The system bay has a yellow sticker marked VJX on it.

    Bay Name     : DB-1A
    Bay LED state: On
    Bay Location : Tile 88AD

    Bay Name     : DB-2A
    Bay LED state: On
    Bay Location : Near Pole21, floor 2, bld8F
```

The above information is only displayed if an EMC Customer Engineer has created it, and if the array is running Enginuity 5875 or 5876.

Setting VMAX 10K and 40K LED door feature (Enginuity 5876 only)

For Enginuity 5876Q22013 SR and higher on VMAX 10K and 40K arrays, the Identification Blinking Door LED feature can be activated and deactivated for a blinking state mode (`blue_flash` or `slow_blink`) simultaneously on all system bays using the `symcfg set` command. The `-led` option is used to indicate that the system or drive bay LED will be controlled. The `-bay_name` option specifies the system bay or drive bay for control of the LED.

The following syntax is used to set the LED by array ID or bay name:

```
symcfg
    set -led [blue_flash | on] [-sid SymmID]

    set -led [slow_blink | on] [-bay_name Bay Name]
```

The bay name (`-bay_name` *`Bay Name`*) parameter must be specified as follows:

SB-x where x is a system bay drive

DB-xy where x is a drive bay index

where y is the bay side (A, B, C, or D)

If the `on` operand is specified and the `-bay_name` option is not specified, the command sets all system bay and drive bay LEDs to ON.

If the `blue_flash` operand is specified, the command sets all system bay LEDs to Blue flash state mode. Drive bay LEDs do not currently support Blue flash and will not be set to this mode when the `blue_flash` operand is specified.

Note: The `slow_blink` operand may be specified either as an abbreviation with a minimum of four characters, `slow` up to the complete `slow_blink`, provided that all supplied characters match. Also, the `blue_flash` operand may be specified either as an abbreviation with a minimum of four characters, `blue` up to the complete `blue_flash`, provided that all supplied characters match.

The following example command sets the LED mode to `blue_flash` for array 188:

```
symcfg set -led blue_flash -sid 188
```

If successful, this command displays:

```
LED state successfully changed.
```

If failed, this command displays a descriptive error message.

The following example command sets the LED mode to `slow_blink` for system bay 2 on array 188:

```
symcfg set -led slow_blink -bay_name SB-2 -sid 188
```

If successful, this command displays:

```
Bay SB-2 LED successfully set to slow blink.
```

If failed, this command displays a descriptive error message.

Note: Controlling the LED door feature requires Enginuity 5876 and higher, and a VMAX 10K or 40K array. This feature is not supported on VMAX 20K arrays.

Use the `symcfg list -v` command to display the LED state as a composite of all bays.

Listing array bay LED states (Enginuity 5876 or lower)

Use the `symcfg list -bay_info` command to display an array bay LED state. The `-bay_info` option is equivalent to the `-bay_location` option, but is more descriptive of its functionality. In addition to Bay Name and Bay Location, a Bay LED State field has been added.

```
Bay Name     : DB-1A
Bay LED state: On
Bay Location :

Bay Name     : SB-1
Bay LED state: On
Bay Location :

Bay Name     : DB-1D
Bay LED state: On
Bay Location :
```

For arrays running Enginuity 5875 or 5876, the reported Bay LED states are: `BLUE_FLASH`, `FAST_BLINK`, `SLOW_BLINK`, `ON`, `OFF`, or `N/A`. The `symcfg list -v` output displays the LED state as a composite of all bays.

```
symcfg list -sid 341 -v
. . .
Maximum number of hypers per disk    :   512
Composite door LED status            : On
Number of Powerpath Devices          :   0
```

```
. . .
```

For a arrays running Enginuity 5874 or lower, the LED status displays N/A:

```
symcfg list -sid 237 -v
. . .
Maximum number of hypers per disk    :   512
Composite door LED status            : N/A
Number of Powerpath Devices          :   0
. . .
```

Note: The Composite door LED status display requires Enginuity 5875 or above. All other LED state display functionality requires Enginuity 5876 or above.

Monitoring Data at Rest Encryption (Enginuity 5875 and 5876)

For arrays running Enginuity 5875 or 5876, Data at Rest Encryption provides hardware-based, on-array, back-end encryption for VMAX family arrays. Back-end encryption protects your information from unauthorized access when disk drives are removed from the system. Data Encryption provides encryption on the back end using Fibre Channel I/O modules that incorporate AES-XTS 256-bit data-at-rest encryption. These modules encrypt and decrypt data as it is being written to or read from disk. All configured drives are encrypted, including data drives, spares, and drives with no provisioned volumes. In addition, all array disk data is encrypted, including array File System and PowerVault contents.

Data at Rest Encryption supports either an internal embedded key manager, or RSA Data Protection Manager for external, enterprise-grade key management. For external key management, Data at Rest Encryption is qualified for interoperability with the RSA Key Manager Appliance version 2.7 SP1, and the RSA Data Protection Manager (DPM) version 3.1 (appliance).

Note: For more information refer to the *EMC Symmetrix VMAX 20K Series Product Guide* and the *EMC Symmetrix VMAX 40K Series Product Guide*.

By securing data on enterprise storage, Data Encryption ensures that the potential exposure of sensitive data on discarded, re-used, or stolen media is reduced or eliminated. As long as the key used to encrypt the data is secured, encrypted data cannot be read. In addition to protecting against threats related to physical removal of media, this also means that media can readily be repurposed by destroying the encryption key used for securing the data previously stored on that media. In this way, disk rotation, migration and upgrade are secured, without changes to operational procedures.

Data Encryption is compatible with all array system features, allows for encryption of any supported local drive types or volume emulations, and delivers powerful encryption without performance degradation or disruption to existing applications or infrastructure.

Note: All key management is transparent to the storage administrator; no direct control of keys is allowed through Solutions Enabler. Encryption must be turned on by an EMC Customer Engineer during the installation of the array. There is no way to turn encryption on or off using Solutions Enabler.

Listing data encryption status

Solutions Enabler displays whether an array has encryption enabled. If encryption is enabled on the array, the `symcfg -v` display shows this information, as in the following example command and output:

```
symcfg list -sid 343 -v

Symmetrix ID: 000194900343
Time Zone   : Eastern Standard Time

    Product Model                        : VMAX
    Symmetrix ID                         : 000194900343
. . .
    3 Dynamic Mirrors                    : Enabled
    Cache Partitioning                   : Disabled
    IPSec Status                         : Pass Thru
    Allow spare in mirror 4 position     : Disabled
    Disks Service                        : Customer Replaceable
    Symmetrix Data Encryption            : Enabled
. . .
```

If encryption has not been turned on at install time, the `symcfg -v` command displays `Disabled` for the row displaying Symmetrix Data Encryption.

If encryption is not supported on the array, the `symcfg -v` command displays `N/A` for the row displaying Symmetrix Data Encryption.

Identifying key management events

Enginuity generates entries in the array Audit Log for every key management event. To view these events, use the following command, which lists the initial DEK (Data encryption key) configuration information:

```
symaudit list -text -sid 012 -function_class Security

             A U D I T   L O G   D A T A

Symmetrix ID            : *********012

  Record                       Function   Action    Activity
  Number   Date      Time      Class      Code      ID
  -------  --------  --------  ---------- -------   ----------------
             Text
             -----------------------------------------------------
        2  11/30/10  14:50:55  Security   Init      SW_Men_6829
             DARE Local Install.

        3  11/30/10  14:50:55  Security   Create    SW_Men_6829
             New KEK key generated. MUID:
   61A3770EDBEE3154463E15A5FA91006C6DA4F9C14C96653635962DE21F636125

        4  11/30/10  14:50:55  Security   Create    SW_Men_6829
            New DEK key generated. Drive WWN: 20000024b65d871b Location: DA07 A0:00
   MUID: 828547DA8EA4D623F674C1FE659329A0D8487308E7BC77979BABC8554ED9B7BF

        5  11/30/10  14:50:55  Security   Create    SW_Men_6829
            New DEK key generated. Drive WWN: 20000024b65d3606 Location: DA07 A0:02
   MUID: C0D587D1EEBC66196EFB63449E31281B50AB3F6B048440661334056FF18EFD4C

        6  11/30/10  14:50:55  Security   Create    SW_Men_6829
            New DEK key generated. Drive WWN: 20000024b65ddb94 Location: DA07 A0:04
   MUID: 2D518DC19DB2D6856DEC5064653EC59CE8FAEA36D11A2078ECA088589F62A647
```

```
      7  11/30/10  14:50:55  Security   Create    SW_Men_6829
         New DEK key generated. Drive WWN: 20000024b65d9bef Location: DA07 A0:06
MUID: 3738B80DF56F65F3E1526E7F16A2721ACED2774A08A35A8B7ABF1442C665F2F1

      8  11/30/10  14:50:55  Security   Create    SW_Men_6829
         New DEK key generated. Drive WWN: 20000024b65ddb3c Location: DA07 A0:08
MUID: 40932F670A3CF4B15909EA309D3A10E1CDD66ED4D5564F75BD75F96A9CA58040

      9  11/30/10  14:50:55  Security   Create    SW_Men_6829
         New DEK key generated. Drive WWN: 20000024b6864000 Location: DA07 A0:0A
MUID: 99234F816501718AA87B29C525976DF7EE9351DC6CAD5756DB7347345F27E3BB

     10  11/30/10  14:50:55  Security   Create    SW_Men_6829
         New DEK key generated. Drive WWN: 20000024b64e634e Location: DA07 A0:10
MUID: 0FC78B4ECA29D0ACD218A05E9A7CE8020EF8E1981CC14B0CB4E849C8A7180482
```

When a drive replacement results in the destruction of the Data Encryption Key for the original drive and the creation of a DEK for the new drive, these events are logged in the array Audit Log. The following output lists this information for array 012:

symaudit list -text -function_class Security -sid 012

```
             A U D I T   L O G   D A T A

Symmetrix ID            : *********012

  Record                      Function   Action    Activity
  Number   Date      Time      Class      Code      ID
  -------  --------  --------  ---------- -------   ----------------
            Text
            -----------------------------------------------------
      135  11/30/10  10:15:58  Security   Modify    SW_Men_E757
            DEK key deactivated. Drive WWN: 20000024b654edce Location: DA08 C1:3E
MUID: E0252F83955C803D03596CF3B3331CED715FA985A6F99E2F6EC0DF325A47458E

      136  11/30/10  10:15:59  Security   Delete    SW_Men_E757
            DEK key destroyed. Drive WWN: 20000024b654edce Location: DA08 C1:3E
MUID: E0252F83955C803D03596CF3B3331CED715FA985A6F99E2F6EC0DF325A47458E

      142  11/30/10  10:17:49  Security   Create    SW_Men_E757
         New DEK key generated. Drive WWN: 20000024b65dd7ea Location: DA08 C1:3E
MUID: 5EC4CF4E2B4D92C98A9FE9A8A19E527B090D95E75C305DAD21E98ED630C5A467
```

The following output lists audit log data about drives added to array 32:

symaudit list -text -function_class Security -sid 032

```
             A U D I T   L O G   D A T A

Symmetrix ID            : *********032

  Record                      Function   Action    Activity
  Number   Date      Time      Class      Code      ID
  -------  --------  --------  ---------- -------   ----------------
            Text
            -----------------------------------------------------
     2700  11/30/10  15:11:15  Security   Create    SW_Men_7348
         New DEK key generated. Drive WWN: 2000001d385398ee Location: DA09 A0:06
MUID: CD45BE05EA03273C78D75EFD58C3092839E218D57E2F5713A188E446EE1D6106

     2701  11/30/10  15:11:17  Security   Create    SW_Men_7348
         New DEK key generated. Drive WWN: 2000001d385529e3 Location: DA09 A1:05
MUID: 69D7D3124001938A031E3CE39058C6D7F21876A0748F73DB33DBF6CE8E5EE137

     2702  11/30/10  15:11:17  Security   Create    SW_Men_7348
```

```
          New DEK key generated. Drive WWN: 2000001d3853aeb4 Location: DA10 A0:05
  MUID: 2C4722A89D473B9AAE70B5706C018B71E231913D76E32C8A8F786A4312FE19CE

    2703  11/30/10  15:11:16  Security   Create    SW_Men_7348
          New DEK key generated. Drive WWN: 2000001d3853a3a8 Location: DA10 A1:06
  MUID: 3FBAEA60781E6E4D1DF31DCD660FE6C6B1C8708FCF85E4B1FAC323DA10DC9207

    2704  11/30/10  15:11:16  Security   Create    SW_Men_7348
          New DEK key generated. Drive WWN: 2000001d3853a53e Location: DA09 B0:05
  MUID: 7518CC4B24D2238F8B93612672B7B6AD014DEDC8CC80FF232F2E06053A9F5CF9

    2705  11/30/10  15:11:17  Security   Create    SW_Men_7348
          New DEK key generated. Drive WWN: 2000001d3853632f Location: DA09 B1:06
  MUID: CFC4658191395836B45105FDD93A3A4CDB6849220164F4D86740E4E9E57A37B4

    2706  11/30/10  15:11:18  Security   Create    SW_Men_7348
          New DEK key generated. Drive WWN: 2000001d3853c06f Location: DA10 B0:06
  MUID: 4FE34F06EF936D115391D49CEDB01EC83396B074CF01001E4CB804D11ADD10E4

    2707  11/30/10  15:11:17  Security   Create    SW_Men_7348
          New DEK key generated. Drive WWN: 2000001d3853af15 Location: DA10 B1:05
  MUID: 6700CBBEF94147353A9E8E63E6A2CC7A53C4116B9C2EB224EEDE49CD59D13610

    2708  11/30/10  15:11:18  Security   Create    SW_Men_7348
          New DEK key generated. Drive WWN: 2000001d3853a932 Location: DA09 C0:06
  MUID: 001874E51C67A16734DC2936E0993178765A2DE5BE6DA3297797E7F7F63B0606

    2709  11/30/10  15:11:18  Security   Create    SW_Men_7348
          New DEK key generated. Drive WWN: 2000001d3853b1e9 Location: DA09 C1:05
  MUID: 8FA8A2094CE7FDC793E5663C28CEE9E1021BF93445DAC484FE129B87357F931A

    2710  11/30/10  15:11:17  Security   Create    SW_Men_7348
          New DEK key generated. Drive WWN: 2000001d385396e4 Location: DA10 C0:05
  MUID: C5DA6495303F86971DEC724D379E58E3E21A6205527C76A40A3BB553A78CCD3E

    2711  11/30/10  15:11:18  Security   Create    SW_Men_7348
          New DEK key generated. Drive WWN: 2000001d3853a4f5 Location: DA10 C1:06
  MUID: 408C812499CA22A0C0375169EA84D456779C7615FAB08FCFA6049FF0F272CEDF
```

Viewing application registrations with array access

You can view all the applications registered with SYMAPI that have accessed all or a specific array to which your host is connected.

For example, to list all the applications that have worked with array `282`, enter `-sid` *`SymmID`* (omit the `-sid` option in order to list connections for all arrays.

```
symcfg list -applications -sid 282

Symmetrix ID     : 000192600282

            Host                                Application
----------------------------  ------------------------------------------------
Node Name
 IP Address
                              ID               Vendor ID        Version   Attr
----------------------------  ---------------- ---------------- --------------

HK192600282
 0.0.0.1
                              OPTIMIZER        EMC Corp         9.1.0.56    -
                              SYMACCESS        EMC Corp         7.3.0.194   -
                              OPTIMIZER        EMC Corp         9.1.0.57    -
                              SMC              EMC Corp         7.3.12.9    -
                              EVTdaemon        EMC Corp         7.3.0.194   -
```

```
                                SMBASE          EMC Corp        7.3.12.9    -
                                SYMACCESS       EMC Corp        7.3.0.177   -
                                OPTIMIZER       EMC Corp        9.1.0.54    -
                                OPTIMIZER       EMC Corp        9.1.5.51    -
                                SMC             EMC Corp        7.3.11.8    -
                                EVTdaemon       EMC Corp        7.3.0.177   -
                                SMBASE          EMC Corp        7.3.11.8    -
                                EVTdaemon       EMC Corp        7.3.0.175   -
                                SYMLMF          EMC Corp        7.3.1202.0  -
                                SYMACCESS       EMC Corp        7.3.0.175   -
                                OPTIMIZER       EMC Corp        9.1.0.46    -

HK192604045
 0.0.0.1
                                OPTIMIZER       EMC Corp        9.1.4.35   (R)

api101
 10.247.80.101
                                SYMCFG          EMC Corp        7.3.0.234  (R)
                                SYMRDF          EMC Corp        7.3.0.230  (M)
                                SYMCG           EMC Corp        7.3.0.230  (M)
                                GNSdaemon       EMC Corp        7.3.0.230  (R)
                                SYMCFG          EMC Corp        7.3.0.230  (R)
                                SYMCG           EMC Corp        7.3.0.227  (M)
                                SYMRDF          EMC Corp        7.3.0.227  (M)
                                SYMCFG          EMC Corp        7.3.0.227  (R)
                                GNSdaemon       EMC Corp        7.3.0.227  (R)
                                GNSdaemon       EMC Corp        7.3.0.228  (R)
                                SYMCFG          EMC Corp        7.3.0.226  (R)
                                GNSdaemon       EMC Corp        7.3.0.224  (R)
                                SYMCFG          EMC Corp        7.3.0.224  (R)
                                SYMCFG          EMC Corp        7.3.0.193  (R)
                                SYMRDF          EMC Corp        7.3.0.189  (M)
                                SYMCFG          EMC Corp        7.3.0.189  (R)

Legend for Attribute(s):

(M) : Application Registered Remotely via RDF links (Multi-Hop).
(R) : Application Registered Remotely via RDF links (One-Hop).
```

Listing host connections to arrays

The `-connections` option of the `symcfg list` command returns the host connections to the array. Only hosts that have at least one registered application are listed. For example, to list the host connections to array 097 enter:

```
symcfg list -connections -capacity -sid 097
```

```
    Host                Symmetrix                      Capacity (GB)
------------  -------------------------  -----------------------------------
Node Name     ID           Director Port  Mapped STDs Mapped BCVs Paired BCVs
------------  ------------ -------- ----  ----------- ----------- -----------

6I            000190100097 FA-3B     0          135.0        67.4         8.4
                                             ----------- ----------- -----------
6I totals:                                       130.8        54.8         8.4

LBQA0074      000190100097 FA-3B     0          135.0        67.4         8.4
                                             ----------- ----------- -----------
LBQA0074 totals:                                   0.0         0.0         0.0

api105        000190100097 FA-3A     0          134.9         0.0         0.0
                                             ----------- ----------- -----------
api105 totals:                                   134.9         0.0         0.0
```

```
api150        000190100097 FA-3A     0          134.9         0.0         0.0
                                               ----------- ----------- -----------
api150 totals:                                   134.9         0.0         0.0

api31         000190100097 FA-7A     0          404.6         0.0         0.0
                                               ----------- ----------- -----------
api31 totals:                                    404.6         0.0         0.0

lbqa0074      000190100097 FA-3B     0          135.0        67.4         8.4
                                               ----------- ----------- -----------
lbqa0074 totals:                                   0.0         0.0         0.0
```

In the previous example, the display of the host connections are sorted according to each array.

Note: To view the host connections for all arrays, omit the `-sid` value.

For IPv6 addresses or host names, which would appear truncated in the default display layout, specify the `-ipv6` option to display a layout that does not truncate node names or addresses. For example, to list all of the host connections (sorted by host names) for array 097, enter:

```
symcfg list -connections -sorthost -sid 097

Host                                   Symmetrix
-----------------------------------------------  --------------------------
Node Name    IP Address      OS Name  OS Revision  ID           Director Port
------------ --------------- -------- -----------  ------------ -------- ----

6I           172.23.195.65   WinNT    5.2.3790     000190100097 FA-3B     0

LBQA0074     172.23.195.74   WinNT    5.2.3790     000190100097 FA-3B     0

api105       172.23.192.105  SunOS    5.8          000190100097 FA-3A     0

api150       172.23.192.150  WinNT    5.2.3790     000190100097 FA-3A     0

api31        172.23.192.31   SunOS    5.8          000190100097 FA-7A     0
```

Listing director configuration data

You can use the `symcfg list` command to gather information about the array directors. The following director types are supported:

- DA — Disk directors
- DX — External disk directors
- EA — ESCON directors
- SE — Gig-E directors
- EF — FICON (Fibre-ESCON) directors
- RA — SRDF directors
- RE — RDF Gig-E directors
- RF — RDF Fibre directors
- FA — Front-end (Fibre Channel) directors, including Fibre Channel over Ethernet (FCoE)
- IM — Infrastructure Manager (IM) directors
- EDS — Enginuity Data Services (EDS) directors

You can use the `symcfg list` command to display information about directors on an array.

For example, to list configuration and status information about all directors on array 00019490004, enter:

```
symcfg list -dir ALL -sid 064
Symmetrix ID: 000197100064 (Local)

    S Y M M E T R I X    D I R E C T O R S

    Ident  Type          Engine  Cores  Ports  Status
    -----  ------------  ------  -----  -----  ------

    IM-7A  IM               1      1      0    Online
    IM-8A  IM               1      1      0    Online
    ED-7B  EDS              2      2      0    Online
    ED-8B  EDS              2      2      0    Online
    DF-7C  DISK             3      2      4    Online
    DF-8C  DISK             3      2      4    Online
    FA-7E  FibreChannel     4      1      2    Online
    FA-8E  FibreChannel     4      1      3    Online
    RF-7G  RDF_BI_DIR       5      1      1    Online
    RF-8G  RDF_BI_DIR       5      1      1    Online
```

Note: Director types IM and EDS are only supported by HYPERMAX OS 5977 or higher.

Support for multiple cores and ports

For arrays running Enginuity 5876 and earlier, there is a fixed relationship between a front-end “emulation instance” (FA, DA, RA, EF, etc.) and its ports. Each emulation instance supports only 1 or 2 ports. Multiple emulations of the same type are allowed on a single director board and these emulations are statically allocated to processors and ports. If you need more IOPS or front-end connectivity, you need to add another emulation instance and its one or two additional ports.

For arrays running HYPERMAX OS 5977, all director types are capable of supporting multiple cores. The actual number of cores assigned to a director are statically configured. You can view the number of cores assigned to a given director using the `symcfg list -dir` command. In addition, with HYPERMAX OS 5977, all director types are capable of supporting a variable number of ports. Solutions Enabler version 8.0.1 includes support for reporting existing port associations, as well as the number and identity of ports available for association (free ports) and their respective supported interface types. A port's supported interface types determine which directors a given free port may be associated with.

Note: To support arrays running HYPERMAX OS 5977, the `symcfg list -dir` command no longer supports the Symbolic, Numeric, and Slot reporting, and has added reporting for engine number, number of cores, and port number.

For example, to associate port 2 to emulation type FA on director 7E, issue the following command:

```
associate port 2 to dir 7E ;
```

After successfully associating a port to a given emulation on a director, you can use the `symcfg list -dir ALL` output to verify that the port is assigned to the requested director as follows:

```
symcfg list -dir ALL -sid 064
```

```
Symmetrix ID: 000197100064 (Local)

    S Y M M E T R I X    D I R E C T O R S

    Ident  Type          Cores  Ports  Status
    -----  ----          -----  -----  ------
    IM-7A  IM              1      0    Online
    IM-8A  IM              1      0    Online
    ED-7B  EDS             2      0    Online
    ED-8B  EDS             2      0    Online
    DF-7C  DISK            2      4    Online
    DF-8C  DISK            2      4    Online
    FA-7E  FibreChannel    1      2    Offline
    FA-8E  FibreChannel    1      3    Online
    RF-7G  RDF_BI_DIR      1      1    Online
    RF-8G  RDF_BI_DIR      1      1    Online
```

Note: Director types IM and EDS are only supported by HYPERMAX OS 5977 or higher.

To disassociate port 2 from emulation type FA on director 7E, issue the following command:

```
disassociate port 2 from dir 7E;
```

After successfully dissociating a port from a given director emulation, you can use the `symcfg list -port -free all` output to verify the association of ports to director emulations. Port 2 is no longer assigned (associated) with any director emulation.

```
symcfg list -port -free -all -sid 064
```

```
Symmetrix ID: 000197100064

              Flags  Speed
  Slot  Port  FCIS   Gb/sec  Status
  ----  ----  -----  ------  -------

     7     2  Y...        8  Powered
     7    18  Y...        8  Powered
     7    20  .Y..        8  Powered
     8    19  ..Y.        8  Powered
     8    21  ...Y        8  Powered

Legend:
  Flags:
    (F)C       : Y = Yes, . = No
    F(C)OE     : Y = Yes, . = No
    F(I)CON    : Y = Yes, . = No
    I(S)CSI    : Y = Yes, . = No
```

Director information by type

If you specify a director type (option `-DA` for example) with the `symcfg` command, the output is filtered to only include records of that type.

Note: Starting with Solutions Enabler V8.0.1, two previously supported director types, `-CA` and `-SA`, have been deprecated. Special format displays such as those previously produced by the specification of `-FA` or `-SE` options are no longer supported. Two new director types have been introduced for Infrastructure Manager (IM) and Enginuity Data Services (EDS).

You can get configuration and status information about one or all directors of a certain type. For example, the following lists all front-end directors (-FA) for array 005:

```
symcfg list -FA ALL -sid 005

Symmetrix ID: 000197300005 (Local)

        S Y M M E T R I X    D I R E C T O R S

    Ident  Type          Engine  Cores  Ports  Status
    -----  ------------  ------  -----  -----  ------

    FA-1E  FibreChannel     1      6      5    Online
    FA-2E  FibreChannel     1      4      4    Online
    FA-3E  FibreChannel     2      6      5    Online
    FA-4E  FibreChannel     2      3      2    Online
```

Director details by name and type

Use the -v (verbose) option to retrieve details about a specific director and type. For example, to display information about the Gig-E director 5G on array 064, enter:

```
symcfg list -SE 5G -sid 064 -v

Symmetrix ID: 000197100064 (Local)
Time Zone   : EDT

    Product Model                        : VMAX40K
    Symmetrix ID                         : 000195700064

      . . .

    Director Identification: SE-5G

        Director Type                    : GigE
        Director Status                  : Online

        Director Symbolic Number         : 05G
        Director Numeric Number          : 101
        Director Engine Number           :  1
        Director Slot Number             :  5

        Number of Director Cores         :  2
        Number of Director Ports         :  1

        Director Port: 16

          WWN Port Name                  : 5000097300039118
          Director Port Status           : Online
          Negotiated Speed (Gb/Second)   :   1
          Director Port Speed (Gb/Second)  :   4
          . . .
```

Director port data

Use the `-port` option to see which ports are online or offline on SA, FA, SE, or RA directors. The `-port` option may be further qualified by `-p <#>` to restrict output to a specific port number.

For example, to get the port status for all SE directors on array 230, enter:

```
symcfg list -port -SE ALL -sid 230 -detail

Symmetrix ID: 000197100230
```

```
      S Y M M E T R I X   D I R E C T O R   P O R T S

                                                  Flags  Speed
    Ident  Port  WWN               Type            ACR   Gb/sec  Status
    -----  ----  ----------------  ------------  -----  ------  -------

    SE-9G     0  50000972C011C918  GigE            X--        8  Online
    SE-10G    0  50000972C011C919  GigE            X--        8  Online
    SE-9H     0  50000972C011C91A  GigE            X--        8  Online
    SE-10H    0  50000972C011C91B  GigE            X--        8  Online

Legend:
  Flags:
      (A)CLX Enabled              : X = True, . = False, - = N/A
      (C)HAP Enabled              : X = True, . = False, - = N/A
      (R)ADIUS Enabled            : X = True, . = False, - = N/A
  . . .
```

Addresses of devices mapped to directors

You can use the `-address` option to identify the address information for devices accessible through specific directors via a host-based view of the storage environment.

For example, to list the address information for all director types for array `064`, enter:

```
symcfg list -dir ALL -sid 064 -address

Symmetrix ID: 000197100064 (Local)

      Director            Device Name          Attr     Address
    ------------ ----------------------------- ---- --------------
    Ident   Port Sym   Physical                     VBUS  TID  LUN
    ------  ---- ----  -----------------------      ----  ---  ---

    FA-8F     10 807A  Not Visible                    0   00  000
                 807B  Not Visible                    0   00  001
                 807C  Not Visible                    0   00  002
                 807D  Not Visible                    0   00  003
```

For arrays running HYPERMAX OS 5977, when LUN information is not available, VBUS TID, and LUN address are reported as N/A.

You can specify the `-p` option to restrict output to a particular port number. For example, to request address information for port 10 on director FA 2E, enter:

```
symcfg list -sid 075 -FA 2E -address -p 10

Symmetrix ID: 000197100075 (Local)

      Director             Device Name          Attr     Address
    ------------ ----------------------------- ---- --------------
    Ident   Port Sym    Physical                     VBUS  TID  LUN
    ------  ---- -----  -----------------------      ----  ---  ---
    FA-2E     10 000AC  Not Visible             ACLX    0   00  100
                 00040  Not Visible                   N/A  N/A  N/A
                 00041  Not Visible                   N/A  N/A  N/A
                 00042  Not Visible                   N/A  N/A  N/A
    Total                             -----
    Mapped Devices:                       4
    Including Metamembers:                4
    Port Available Addresses:          4092
    Director Available Addresses:    98284 (s)

Legend for Available address:
```

```
(s): The Available Addresses for a director are shared among
     its ports (shared)
```

Next available device address

You can specify the `-availabile` option with `-address` to return the next available address that can be used for a device. VBUS, TID, and LUN address values with an asterisk (*) represent a gap in the address assignments, or are the next available address in the run. For example, to list the available address information, enter:

```
symcfg list -dir all -address -available -sid 097

Symmetrix ID: 000190100097

          Director                  Device Name          Attr     Address
---------------------- ----------------------------- ---- -------------
Ident   Symbolic  Port Sym   Physical                      VBUS  TID  LUN
------  --------  ---- ----  -----------------------      ----  ---  ---

FA-3A   03A       0    0E2A  /dev/vx/rdmp/c5t0d0s2           0   00  000
                       0E2B  /dev/vx/rdmp/c5t0d1s2           0   00  001
                       04AA  /dev/vx/rdmp/c5t0d2s2   (M)     0   00  002
                       04AE  /dev/vx/rdmp/c5t0d3s2   (M)     0   00  003
                       04B2  /dev/vx/rdmp/c5t0d4s2   (M)     0   00  004
                       04B6  /dev/vx/rdmp/c5t0d5s2   (M)     0   00  005
                       04BA  /dev/vx/rdmp/c5t0d6s2   (M)     0   00  006
                       04BE  /dev/vx/rdmp/c5t0d7s2   (M)     0   00  007
                       04C2  /dev/vx/rdmp/c5t0d8s2   (M)     0   00  008
                       04C6  /dev/vx/rdmp/c5t0d9s2   (M)     0   00  009
                       -     AVAILABLE                       0   00  00A *
```

Taking RA directors offline

One or all RA directors of a specified array can be taken offline.

For example, to take `RA-12` of array `097` offline, enter:

```
symcfg offline -RA 12 -sid 097
```

To bring the director back online, enter:

```
symcfg online -RA 12 -sid 097
```

Taking front-end director ports offline

A specific port of a front-end director can be taken offline as well.

For example, to take port `1` of `SA-12` in array `097` offline, enter:

```
symcfg offline -SA 12 -P 1 -sid 097
```

CAUTION

If you turn off the only connection from your host to your array array, you will not be able to contact the array in order to turn it back on (online).

If you have turned off the only connection from your host, you can use another host, if one exists, to bring the director port back online. For example, to bring port `1` of `SA-12` in array `097` back online, enter:

```
symcfg online -SA 12 -P 1 -sid 097
```

Array external locks

Array external locks are used by SYMAPI (locks 0 to 15) and also for applications assigned by EMC (>15) to lock access to the entire array during critical operations. (Base SRDF operations use lock 0 and the Optimizer uses lock 13.) Using the `symcfg list -lockn` parameters, you can list all locks on one or all arrays or just those locks targeted to specific operations.

A list can be returned for all, a specific number, only SRDF, only SRDF-A, only MSCS, or only GNS locks (`ALL`, `#`, `RDF`, `RDFA`, `SRDF_MSCS`, and `GNS` values respectively).

Listing all locks

To return a list of all host-visible arrays (local and remote), along with details about all array exclusive locks (known to SYMAPI) that are currently locked, enter:

```
symcfg list -lockn all

             S Y M M E T R I X      L O C K S

                                Lock       Lock    Lock                Time
    SymmID       Attachment     Status     Number  Usage               Held (Sec)

    000000006196 Local          Unknown    N/A     N/A                 N/A
    000184600063 Local          Unknown    N/A     N/A                 N/A
    000184600282 Local          Unknown    N/A     N/A                 N/A
    000187900039 Remote         EXCLUSIVE  15      ConfigChg           307
```

In the previous example, the returned list contains three local arrays that have no known locks, as specified by the N/A values. However, remote array 000187900039 has an exclusive lock number 15 for a configuration change activity.

Listing details about a lock number

To focus your returned data on one or all lock number(s) on one or all array(s), use the `-lockn` option. For example, to return a list of all host-visible arrays (local and remote), along with details about any lock 0, enter:

```
symcfg list -lockn 0

                        S Y M M E T R I X      L O C K S
                                Lock       Lock    Lock                Time
    SymmID       Attachment     Status     Number  Usage               Held (Sec)

    000000006196 Local          Unknown    N/A     N/A                 N/A
    000184600063 Local          Unknown    N/A     N/A                 N/A
    000184600282 Local          Unknown    N/A     N/A                 N/A
    000187900039 Remote         Unknown    N/A     N/A                 N/A
```

In the example above, the returned list shows that none of the arrays currently have lock 0 locked, as specified by N/A.

Note that array 000187900039 still holds lock 15 from the previous example, however, the command line query specifically requested information on exclusive lock 0, so the returned data for a array that held any other lock number would be returned as N/A.

Note: To return information on all known array exclusive locks currently locked, use the keyword `ALL` with the `-lockn` option. When using `symcfg list -lock`, where the `-lockn` option is omitted, the lock number 0 (base SRDF operations) is assumed.

Listing details about a lock

Using the `-lockn` option with list, you can view whether one or more Symmetrix arrays have an external lock held. By default, only lock 0 is checked. For specific locks use the `-lockn` option with the following keywords:

- `-lockn ALL` command checks for all Symmetrix external locks known to SYMAPI.
- `-lockn RDF` command checks for locks specific to RDF.
- `-lockn GNS` command lists the GNS specific locks.
- `-lockn SRDF_MSCS` lists the SRDF/CE for MSCS specific locks.

In addition, you can specify a specific lock number.

For example, to return information about lock 23, enter:

```
symcfg list -lockn 23  -sid 190

Symmetrix ID: 000000006190

               S Y M M E T R I X      L O C K S

                                Lock       Lock    Lock                Time
    SymmID       Attachment     Status     Number  Usage               Held (Sec)
    000000006190 Local          Locked     23      Unknown             20
```

In addition, the verbose option (`-v`) can be used to provide, if available from the base daemon, extended lock information about the application and host that owns the lock. For example, to return verbose information about lock 15, enter:

```
symcfg list -lockn 15 -sid 207 -v

Symmetrix ID: 000192600207

    Symmetrix ID                        : 000192600207   (local)
    Lock Number                         : 15
    Lock Usage                          : Config Change
    Time Held in Seconds                : 1801
    Lock Holder ID                      : 0xf000128e

    Initiator                           : 0
    Director Number                     : 55
    Logical Path ID                     : -1
    Lock Owner                          : Config Server
    Config Change Session ID            : 4356
    Migration Session Name              : N/A
```

Releasing a lock

SYMCLI provides the ability to release a lock on a array. This is not a recommended procedure and is only useful for locks which you have confirmed are stranded. For example, to release an external lock `0`, which has confirmed to be stranded on array `097`, enter:

```
symcfg -sid 097 -lockn 0 release
```

When Solutions Enabler is run against an array running Enginuity 5874 or higher, the `symcfg release` command determines the lock owner and releases the lock. If held by an SRDF control operation, or held by the configuration change server, SYMCLI blocks the user command and displays a failure message identifying the lock owner. The lock owner can be a configuration change session, a migrate session, or an internal change. The user can then abort the correct session that owns the lock, using the `symconfigure` or the `symmigrate` command, whichever is applicable.

For information about device external locks that only target specific devices, refer to "Releasing device external locks" on page 474.

LRU cache management group

All, or a specific, least-recently-used (LRU) cache management groups for one or all arrays can be obtained using the `-lru` option of the `symcfg list` command. For example, to view a list of all LRUs for the array `6196`, enter:

```
symcfg list -lru all -sid 6196

Symmetrix ID: 000000006196 (Local)

    LRU Num   LRU Name    Cache Slots  Percent of Total
    -------   --------    -----------  ----------------
          1    GROUP_0          11971        4%
          2    GROUP_1          11971        4%
          3    GROUP_2          11971        4%
          4    GROUP_3          11971        4%
          5    GROUP_4          11971        4%
          6    GROUP_5          11971        4%
          7    GROUP_6          11971        4%
          8    GROUP_7          11971        4%
          9    GROUP_8          11971        4%
         10    GROUP_9          11971        4%
         11    GROUP_A          11971        4%
         12    GROUP_B          11971        4%
         13    GROUP_C          11971        4%
         14    GROUP_D          11971        4%
         15    GROUP_E          11971        4%
         16    GROUP_F          11971        4%
                 -------    ---------
                   Total       278616
```

The output returned includes a table listing the LRU number and name, the cache slots that LRU occupies, and the percentage of the total cache. In addition, the total number of cache slots occupied by the listed LRUs is provided at the bottom.

Optimized Read Miss mode (Enginuity 5876 only)

The Optimized Read Miss feature reduces the I/O processing overhead of read miss operations for both DA and DX emulations. The feature is supported on arrays running Enginuity 5876Q22013 or any higher version of 5876. Solutions Enabler reports the current Optimized Read Miss state and permits setting the Optimized Read Miss mode to On or Off for a device, a list of devices, or devices in a storage group, device group, or composite group. The Off setting explicitly disables the Optimized Read Miss mode for the specified devices. The System setting which is the default mode, allows the array to determine the appropriate Optimized Read Miss mode for the specified devices. Optimized Read Miss mode is supported only for FBA devices or AS400 devices with D910 emulation.

Note: All members of a meta device should have the same Optimized Read Miss mode. The `symconfigure` command used for creating and modifying a meta device prevents building a meta device with different settings. In the event of inconsistent modes for meta members, the Optimized Read Miss mode of the head device is reported for the meta head.

Use the `symdev set -orm` command to set the Optimized Read Miss mode to On or Off, or System managed for a single device or a list of devices. Use the `symdev list` command to indicate whether Optimized Read Miss mode is currently set to On, Off, or System managed for a device. In addition, output can be filtered by using the `-orm off` option to select devices with Optimized Read Miss mode set to Off.

Use the following `symdev` command syntax to set the Optimized Read Miss mode for a single device or a list of devices:

```
symdev -sid SymmID [-devs <<SymDevStart>:<SymDevEnd> | <SymDevName>
              [,<<SymDevStart>:<SymDevEnd> | <SymDevName>>...]>]
[-noprompt] [-rp] [-celerra] [-star]

       set -orm [on | off | system]


symdev -sid SymmID -file FileName
       [-noprompt] [-celerra] [-star]

       set -orm [on | off | system]
```

Output from the `symdev show` command indicates whether Optimized Read Miss mode is currently disabled (Off), or system managed (System) for a device. If the ORM feature is not available for your Enginuity version, the Optimized Read Miss mode displays as N/A.

For example:

```
symdev show 01F2 -sid 46
    . . .
    AS400_GK                 : False
    Host Cache Registered    : True
    Optimized Read Miss      : Off

    Mirror Set Type          : [RAID-1,N/A,N/A,N/A]
    . . .
```

Note: The Optimized Read Miss mode can also be set to Off or System Managed for devices in a storage group (`symsg`), device group (`symdg`), or composite group (`symcg`).

Network services

The Solutions Enabler configuration file `netcnfg` is used by the SYMCLI client library to specify attributes of a remote server where array management operations need to be directed. Each line in the file associates a user-chosen service name with host name or IP address, port, and security level information.

The `netcnfg` file is a template and an editable file located in the SYMAPI configuration directory. The location of this directory varies according to operating system. The *Solutions Enabler Installation Guide* explains how to set up a Solutions Enabler client and configure services in the `netcnfg` file.

The `symcfg -services list` command validates the syntax of the file entries and then displays the configured network services available for use by the SYMAPI client connection. For example:

```
symcfg list -services

                    S Y M A P I   N E T   S E R V I C E S

             Pairing                                                  Port  Security
  Name        Method   Type     Node Name                Address      Number Level
------------ ------   -------- -------------------- -------------------------- ----------
single_entry Single   TCPIP    host1.xyz.com                                2707 SECURE
ord_host     Ordered  TCPIP    host1.xyz.com                                2707 SECURE
ord_host     Ordered  TCPIP    host2.xyz.com                                2707 SECURE
bal_host     Balanced TCPIP    host1.xyz.com                                2707 SECURE
bal_host     Balanced TCPIP    host2.xyz.com
```

Memory board information

SYMCLI provides the ability to view memory board information per array. This information includes the number of boards, the slot number, and the capacity information in MBs. To view the available memory board information for all arrays, enter:

```
symcfg list -memory
```

Mainframe CU image

The `symcfg` command can be used to return mainframe CU (Controller Unit) image information if the storage environment contains devices mapped to either EA (ESCON) or EF (FICON) front-end directors. Since devices in the mainframe environment are managed with respect to the CU image that they are a part of, SYMCLI has created a view of the CU images that are defined within the array. A CU image definition includes the SSID assigned to the image, the front-end ports to which it is mapped, the devices included in the image, and their base and alias addresses. It also indicates whether it uses dynamic or static PAV (parallel access volumes) and whether the CU is online or not.

To view a list of CU images for all arrays (you can optionally limit the return data to one array by specifying the array ID), use the following command:

```
symcfg list -cuimage
```

From the list of returned CU images, you can obtain details about a specific image using the `show` command:

```
symcfg show -cuimage 0 -ssid_num 640
```

The example above makes use of the `-ssid_num` option to return the specified CU image located on a specific subsystem ID in the case where multiple CU images with the number `0` may exist across multiple subsystems.

Enginuity patches

To list all of the Enginuity patches installed on a specific array, enter:

```
symcfg list -upatches -sid 207

Symmetrix ID: 000192600207

 Enginuity Level    : 5874_63
```

```
Total Patches      : 2371
Permanent Patches  : 2347
Temporary Patches  : 24

P00022136:001 P00033135:001 P00033563:012 P00034999:002 P00035237:001
P00035422:001 P00035530:009 P00036633:001 P00036812:001 P00036889:002
P00036896:001 P00037223:004 P00037353:001 P00037388:001 P00037536:001
P00037577:001 P00037580:001 P00037648:001 P00037747:001 P00037817:002
P00037894:001 P00038171:001 P00038219:001 P00038287:001 P00038326:001
P00038347:001 P00038352:001 P00038403:001 P00038485:001 P00038612:006
P00038713:001 P00038785:001 P00038821:001 P00038896:004 P00038921:007
P00038967:001 P00039002:002 P00039013:001 P00039089:001 P00039111:002
T00039113:002 P00039146:001 P00039151:001 P00039195:001 P00039208:001
P00039224:001 P00039289:001 P00039300:001 P00039319:001 P00039338:001
. . .
```

Environmental data

The status of the major hardware modules including fans, power supplies, drive enclosures, and link control cards can be returned by specifying the `list -env_data` option. When querying for environmental data, an array ID must be specified.

Use the following command syntax to return an overall status for all enviromental components of a specified array:

`symcfg -sid` *`SymmID`* `list -env_data`

For example, to return an overall status for all environmental components of a given array, enter:

```
symcfg -sid 150 list -env_data
```

For example, to return a detailed status for each environmental component of a given bay (in this example, the bay name is `SystemBay`) for a specified array, enter:

```
symcfg -sid 150 show -env_data SystemBay
```

The `symcfg` command provides options that allow the returned data to be limited to a specific service state (`-service_state` option) — either `degraded`, `failed`, or `normal`—, or all service states except one by preceding the service state value with a `not`, such as `-service_state notfailed`.

For example, to return the environmental data for array 150 with a service state of `failed`, enter:

```
symcfg -sid 150 list -env_data -service_state failed
```

Note: The returned data only contains information about the bay containing the failure, and the components in the failed state.

Listing array environmental data

The following example lists environmental output for an array running HYPERMAX OS 5977 containing a system bay and at least one drive bay:

```
symcfg list -env_data -v -sid 64
```

```
Symmetrix ID               : 000195700064
Timestamp of Status Data   : 09/21/2011 14:01:51

    System Bays
```

```
Bay Name                                :    SB-1
      Bay LED state                      :    Normal  (On)
      Front Door Bay LED state           :    Normal  (On)
      Rear Door Bay LED state            :    Normal
      Number of Standby Power Supplies   :    2
      Number of Drive Enclosures         :    1
      Number of Enclosure Slots          :    1
      Number of MIBE Enclosures          :    2

      Status of Contained Modules
        Standby Power Supplies
          SPS-1A (Aggregate)             :    Normal
            SPS-TRAY-1A                  :    Normal
            SPS-BATTERY-1A               :    Normal
          SPS-1B (Aggregate)             :    Normal
            SPS-TRAY-1B                  :    Normal
            SPS-BATTERY-1B               :    Normal

        Enclosure Slot Number            :    1
          Enclosure Slot State           :    Normal
          MM-7                           :    Normal
          MM-8                           :    Normal
          DIR-1                          :    Normal
            PS-A                         :    Normal
            FAN-1                        :    Normal
            BOOT-DRIVE-0                 :    Normal
          DIR-2                          :    Normal
            PS-B                         :    Normal
            FAN-2                        :    Normal
            BOOT-DRIVE-0                 :    Normal

        Drive Enclosure Number           :    1
          Drive Enclosure State          :    Normal
          SSC                            :    Normal
          LCC-A                          :    Normal
          LCC-B                          :    Normal
          ICM-A                          :    Normal
          ICM-B                          :    Normal
          PS-A                           :    Normal
          PS-B                           :    Normal
          FAN-1                          :    Normal
          FAN-2                          :    Normal

        MIBE Name                        :    MIBE-A
          MIBE State                     :    Normal
          PS-A                           :    Normal
          PS-B                           :    Normal
          CM                             :    Normal

        MIBE Name                        :    MIBE-B
          MIBE State                     :    Normal
          PS-A                           :    Normal
          PS-B                           :    Normal
          CM                             :    Normal

Drive Bays

      Bay Name                           :    DB-1A
      Bay LED state                      :    Normal (On)
      Number of Standby Power Supplies   :    4
      Number of Drive Enclosures         :    16

      Status of Contained Modules
        Standby Power Supplies
          SPS-1A (Aggregate)             :    Normal
            SPS-TRAY-1A                  :    Normal
```

```
      SPS-BATTERY-1A                    :    Normal
    SPS-1B                              :    Normal
    SPS-4A                              :    Normal
    SPS-4B                              :    Normal

  Enclosure Number                      :    2
    Enclosure State                     :    Normal
    MM-A                                :    Normal
    MM-B                                :    Normal
    DIR-3                               :    Normal
    DIR-4                               :    Normal
```

Differences in hardware configurations between arrays running HYPERMAX OS 5977 and arrays running Enginuity 5876 or lower requires changes to the Environmental Data reporting in Solutions Enabler 8.0.1. Changes are made to the output display for `symcfg list -env_data -v` command to support these differences. The output now reports as follows:

- Director-specific Power Supplies, Fans and Boot Drives under the Director to which they are dedicated.
- Reporting for more than one Enclosure Slot (Engine) per System Bay.
- Cooling Modules (CM) for MIBE enclosures.
- Dedicated Fans (FAN) for Drive Enclosures (DAEs).
- Inter Connect Module (ICM) for Drive Enclosures (DAEs).
- Status System Cards (SSC) for Drive Enclosures (DAEs).
- Management Module Control Station (MMCS) modules for Enclosure Slots (Engines).
- For System Bays or Drive Bays with both Front and Rear door LEDs, the existing Bay LED state line will contain an aggregate service state (Normal, Degraded or Failed) based on the service states of the Front and Rear door LEDs as follows:
  - If the service state of either LED is N/A, the aggregate state will be N/A.
  - If the service state of either LED is Failed, the aggregate state will be Failed.
  - If the service state of either LED is Degraded, the aggregate state will be Degraded.
  - Otherwise, the aggregate service state will be Normal.
- Information for both Front and Rear door bay LED modules for System Bays and Drive Bays.
- The single Front door bay LED state, for arrays running Enginuity 5876 or lower, is reported on both the new Front Door Bay LED state line and the existing Bay LED state line.
- Enclosure Slots (Engines) are reported as Number of Enclosure Slots and Drive Enclosures (DAEs) are reported as Number of Drive Enclosures. The Enclosure Slots (Engines) and Drive Enclosures (DAEs) are sorted numerically in the display for easier reference.
- For arrays running Enginuity 5876 or higher, for each tray/battery pair, a line representing the aggregation (Dual component standby power supply) of associated tray/battery components is displayed. State information is reported at both the

aggregate and components levels. The module name appears as `SPS-XX (Aggregate)` , where `XX` identifies the unique alpha-numeric identifier of the corresponding tray and battery components.

- The dual-component SPSs are only counted once when calculating the Number of Standby Power Supplies displayed

Note: With Enginuity version 5875 or higher, an environmental module service state of N/A displays when the actual service state cannot currently be obtained.

Device pools

The `symcfg list -pool` command lists information for all pool types. You can limit the display to a certain type of pool by using the options `-snap`, `-rdfa_dse,` or `-thin`. The `-rdfg` *`GrpNum`* option can be used with the `-rdfa_dse` option to limit the display to the SRDF/A DSE pools that are related to a specified SRDF group. When used, the returned data includes pools that are currently associated with the group or pools that have been disassociated from the group but still have the group's data.

Note: When issued against a VMAX 10K array with Enginuity 5875Q32011SR, the `symcfg list -pool` and `symcfg show -pool` commands report information about SRDF/A DSE pools.

```
symcfg  [-sid SymmID] [-offline] [-mb | -gb]
       [-i Interval] [-c Count]

   list [-pool [-snap][-rdfa_dse [-rdfg GrpNum]][-thin]
          [-fba] [-ckd] [-ckd3390] [-ckd3380] [-as400] [-all] [-v]]
```

Where:

`-i Interval -c Count` — The interval and count options can be used to look at the status of the pool(s) continuously for a certain period of time.

-v — used to list details of each pool in the returned data set. It is equivalent to using the `symcfg show -pool` command on all desired pools.

`-mb | -gb | -tb` — By default, the space consumption of devices and pools is shown as a number of tracks. For the `symcfg list` and `symcfg show` commands, the output can be shown in megabytes, gigabytes, or terabytes by specifying one of these options. The gigabytes display has one decimal point precision.

`-pool` — Lists information for TF/Snap, SRDFA/DSE, and thin pools in a common output.

`-snap` — Lists information for TF/Snap pools only.

`-rdfa_dse` — Lists information for SRDF/A DSE pools only.

`-rdfg` — Used with `-rdfa_dse` to limit the display to the SRDF/A DSE pools that are related to the specified SRDF group. This includes pools that are associated with the group and pools that have been disassociated from the group, but may still have some data for the group.

`-thin` — Displays information about Virtual Provisioning ™ thin device pools.

-all — Includes both enabled and disabled devices in the calculations for `Free Tracks` and `Full %`. Otherwise, only enabled devices are included. When you specify the `-all` option, both enabled and disabled devices are included in the calculations of `Free Tracks` and `Full %`. Otherwise, only enabled devices are included. For example, without the `-all` option, the Free Tracks field include free tracks from all enabled devices and the `Full %` field would be based on the Enabled tracks. With the `-all` option specified, the Free Tracks field include free tracks from both enabled and disabled devices and the `Full %` would be based on the Usable Tracks.

-fba | -ckd | -ckd3390 | -ckd3380 | -as400 — Filters the pool display to the specified emulation type.

The display of the pool information for the `symcfg list pool` command includes the pool name, pool type, device emulation type in the pool, tracks usage of the pool as well as pool state. The pool state indicates whether there are any devices enabled in the pool. If yes, pool state is Enabled, otherwise it is Disabled.

The following example shows the details of a thin pool:

```
symcfg show -sid 049  -pool  xx_thin  -thin -detail
Symmetrix ID: 000197100049

Symmetrix ID                     : 000197100049
Pool Name                        : xx_thin
Pool Type                        : Thin
Disk Location                    : Internal
Technology                       : SATA
Dev Emulation                    : FBA
Dev Configuration                : 2-Way Mir
Pool State                       : Enabled
Compression State                : N/A
# of Devices in Pool             :  10
# of Enabled Devices in Pool     :  10
# of Usable Tracks in Pool       : 99000
# of Allocated Tracks in Pool    : 14081
# of Thin Device Tracks          : 11549
# of DSE Tracks                  : 2298
# of Local Replication Tracks    : 234
# of Tracks saved by compression : 0
# of Shared Tracks in Pool       : 0
Pool Utilization (%)             :   3
Pool Compression Ratio (%)       :   0
Max. Subscription Percent        : N/A
Rebalance Variance               : N/A
Max devs per rebalance scan      : N/A
Pool Reserved Capacity           : N/A

Enabled Devices(1):
  {
   ----------------------------------------------------------
   Sym         Usable      Alloc       Free Full FLG Device
   Dev         Tracks     Tracks     Tracks  (%)  S  State
   ----------------------------------------------------------
   02010        33000      11549      21451   35  .  Enabled
   03010        33000       2298      30702    7  .  Enabled
   04010        33000        234      32766    1  .  Enabled
           ---------- ---------- ---------- ----
   Tracks       99000      14081     849219   14
  }

Pool Bound Thin Devices(16):
  {
   ----------------------------------------------------------
```

```
                                Pool          Pool        Compressed
Sym    FLG        Total Sub      Allocated     Size/Ratio
Dev     T        Tracks (%)     Tracks (%)     Tracks (%)
-----------------------------------------------------------
08070   B         33000   5       9000  28       9000   0
08071   B         33000   5       2298   7       2208   0
08072   B         33000   5          0   0          0   0
08073   B         33000   5          0   0          0   0
08074   B         33000   5        251   1        251   0
08075   B         33000   5          0   0          0   0
08076   B         33000   5          0   0          0   0
08077   B         33000   5          0   0          0   0
08078   B         33000   5          0   0          0   0
08079   B         33000   5          0   0          0   0
0807A   B         33000   5          0   0          0   0
0807B   B         33000   5          0   0          0   0
0807C   B         33000   5          0   0          0   0
0807D   B         33000   5          0   0          0   0
0807E   B         33000   5          0   0          0   0
0807F   B         33000   5          0   0          0   0
                ---------- --- ---------- --- ---------- ---
Tracks          528000  80      11549  22      11549   0
}


Legend:
  Enabled devices FLG:
    (S)hared Tracks : X = Shared Tracks , . = No Shared Tracks
  Bound Devices FLG:
    S(T)atus    : B = Bound, I = Binding, U = Unbinding, A = Allocating,
                  D = Deallocating, R = Reclaiming, C = Compressing,
                  N = Uncompressing, F = FreeingAll, . = Unbound,
```

Listing all Virtual Provisioning pools in an array

To list only the Virtual Provisioning pools in the output, use the `-thin` option. The following output lists details about all thin pools in array 087:

symcfg list -pool -sid 087 -thin -all -detail

```
Symmetrix ID: 000197100087

                              S Y M M E T R I X   T H I N   P O O L S
-----------------------------------------------------------------------------------------------------
Pool         Flags  Dev              Total     Usable       Free       Used Full Subs Comp     Shared
Name         PTECSL Config          Tracks     Tracks     Tracks     Tracks  (%)  (%)  (%)     Tracks
------------ ------ ------------ ---------- ---------- ---------- ---------- ---- ---- ---- ----------
DG1_CKD10K   TFC-EI 2-Way Mir       6274800    6274800    6274800          0    0    0    0          0
DG1_FBA10K   TFF-EI 2-Way Mir      11536560   11536560    2469946    9066614   78    0    0          0
DG2_FBA7_2   TSF-EI RAID-5(3+1)    21546000   21546000   21546000          0    0    0    0          0
DG3_FBA7_2   TSF-EI 2-Way Mir       3591000    3591000    3591000          0    0    0    0          0

Total                            ---------- ---------- ---------- ---------- ---- ---- ---- ----------
Tracks                             42948360   42948360   33881746    9066614   21    0    0          0

Legend:
  (P)ool Type:
    S = Snap,  R = Rdfa DSE  T = Thin
  (T)echnology:
    S = SATA, F = Fibre Channel, E = Enterprise Flash Drive, M = Mixed, - = N/A
  Dev (E)mulation:
    F = FBA, A = AS400, 8 = CKD3380, 9 = CKD3390, - = N/A
  (C)ompression:
    E = Enabled, D = Disabled, N = Enabling, S = Disabling, - = N/A
  (S)tate:
    E = Enabled, D = Disabled, B = Balancing
  Disk (L)ocation:
    I = Internal, X = External, M = Mixed, - = N/A
```

Showing Virtual Provisioning pool rebalancing

The `symcfg show -thin -pool -detail -all` report shows the following pool rebalancing parameters:

- **Rebalancing variance** — Targets the device utilization variance for the rebalancing algorithm. The rebalancing algorithm attempts to level distribution of data in a pool so that the percentage utilization of any device in the pool is within the target variance of the percentage utilization of any other device in the pool.
- **Max devs per rebalance scan** — Lists the maximum number of devices in a pool to use in the rebalancing algorithm. The following output shows `256`, which is the default.

Pool rebalancing is for thin pools only and is not applicable to Snap or SRDF/A DSE pools.

```
symcfg show -pool Mig_trg2 -thin -sid 432 -all -detail

Symmetrix ID: 000194900432

Symmetrix ID                     : 000194900432
Pool Name                        : Mig_trg2
Pool Type                        : Thin
Disk Location                    : External
Technology                       : N/A
Dev Emulation                    : FBA
Dev Configuration                : 2-Way Mir
Pool State                       : Enabled
Compression State                : Enabled
# of Devices in Pool             :  10
# of Enabled Devices in Pool     :  10
# of Usable Tracks in Pool       : 99000
# of Allocated Tracks in Pool    : 14081
# of Thin Device Tracks          : 11549
# of DSE Tracks                  : 2298
# of Local Replication Tracks    : 234
# of Tracks saved by compression : 0
# of Shared Tracks in Pool       : 0
Pool Utilization (%)             :   3
Pool Compression Ratio (%)       :   0
Max. Subscription Percent        : N/A
Rebalance Variance               : N/A
Max devs per rebalance scan      : N/A
Pool Reserved Capacity           : N/A

. . .

Legend:
  Enabled devices FLG:
    (S)hared Tracks : X = Shared Tracks , . = No Shared Tracks
  Bound devices FLG:
    S(T)atus    : B = Bound, I = Binding, U = Unbinding, A = Allocating,
                  D = Deallocating, R = Reclaiming, C = Compressing,
                  N = Uncompressing, . = Unbound
```

Note: The **FLG S** flag field indicates whether or not there are shared allocations on each device as noted in the above legend.

Listing feature registrations and usage data

The `symcfg list -feature` command lists feature class registrations and usage data for a specified array. Where appropriate, capacity types and limits are also displayed. Usage information is displayed when the -v option is provided.

You can limit the feature display to a specified class of features by using the `-class` option. The following example lists array feature classes for Local Replication:

```
symcfg list -feature -sid 341 -class Local Replication

Symmetrix ID : 000194900341

    Feature Name           : TimeFinder/Mirror
    Feature Type           : Product
    Feature Class          : Local Replication
    Feature Capacity Type  : TB of Total Capacity
    Feature Capacity       : 0
    SATA Capacity          : 0
    Enabled Status         : Disabled
    Enabled Change Date    : 05-Jan-2011 17:10
    Feature Name           : SYMM_TF_CLONE
    Feature Type           : Product
    Feature Class          : Local Replication
    Feature Capacity Type  : TB of Registered Capacity
    Feature Capacity       : 500
    SATA Capacity          : 0
    Enabled Status         : Disabled
    Enabled Change Date    : 10-Mar-2011 14:32
    Feature Name           : SYMM_TF_SNAP
    Feature Type           : Product
    Feature Class          : Local Replication
    Feature Capacity Type  : TB of Total Capacity
    Feature Capacity       : 100
    SATA Capacity          : 500
    Enabled Status         : Disabled
    Enabled Change Date    : 10-Mar-2011 14:30
```

Note: When issued against a VMAX 10K array with Enginuity 5875Q32011SR, the `symcfg list -feature` CLI command returns a list of additional feature classes that are supported for VMAX 10K.

Device-level data

From the perspective of software running on a host system, an array appears to be a number of *physical devices* connected to one or more I/O controllers. A host application addresses each of these devices using a physical device name. Each physical device defined in the configuration database has a specific set of attributes (such as vendor ID, product ID, revision level, and serial ID).

A device can map to a part of a physical disk or to an entire disk. The part of a physical disk to which a device is mapped is called a *hypervolume* or a *hyper*. A device may map to multiple hypers (containing identical copies of data) depending on its mirror configuration.

The array database file maintains device-level configuration and status information for each device on every array that is accessible from the host. Using SYMCLI, you can obtain a list of all the available devices. You can then use the returned device data from this list

to obtain configuration and status information. This information can lead you to back-end information for the device's disk directors and corresponding hypervolumes, and their mappings to disk drives.

Device types - HYPERMAX OS 5977

SYMCLI provides the ability to define and configure devices for numerous specialized roles defined as *device types*. Each device type has specialized characteristics to enable a device to participate in various SYMAPI operations.Table 28 and Table 29 describe the device types of the DMX and VMAX storage environment.

Table 28 Device types - HYPERMAX OS 5977

Device type	Description
ACLX devices	ACLX devices are device masking devices similar to VCM devices that are used for storage provisioning using auto-provisioning groups. VMAX3 arrays come pre-configured with one ACLX Device. The ACLX attribute will cannot be removed, and the ACLX devices cannot be created or deleted. The device will be visible to hosts at the LUN Address configured in the ACLX Device Lun address global configuration, and zero is the default value. The ACLX device is only visible on front end ports where the `Show_ACLX_Device` port attribute is enabled.
Data Domain devices	An encapsulated Data Domain device.
Gatekeeper devices	Gatekeeper devices are LUNs that act as the target of command requests to microcode-based functionality. For detailed information on gatekeeper management, refer to the *EMC Solutions Enabler Installation Guide*.
RDF devices	Devices configured as RDF1, RDF2, or RDF21 to support SRDF operations. SRDF is a business continuance solution that maintains a device-level mirror of array data on remotely attached arrays. These arrays also may be located in physically separate sites. SRDF provides a recovery solution for component or site failures using remotely mirrored devices. SRDF reduces backup and recovery costs and significantly reduces recovery time after a disaster. For more information, refer to the *EMC Solutions Enabler SRDF CLI Product Guide.*
Thin devices (TDEV)	Thin devices for Virtual Provisioning are devices that do not have storage allocated to them when they are created. To a host operating system, they look like regular devices with their configured capacity. The host treats them as regular devices and writes and reads from these devices like regular devices.
RecoverPoint Splitter devices	Devices that are in use by the RecoverPoint Splitter. With HYPERMAX OS 5977, indicates the specific RecoverPoint device type as production, replica or internal.

Device types - Enginuity 5876 and lower

SYMCLI provides the ability to define and configure devices for numerous specialized roles defined as *device types*. Each device type has specialized characteristics to enable a device to participate in various SYMAPI operations. Table 29 on page 451 describes the device types of the array storage environment.

Table 29 Device types - Enginuity 5876 and lower (page 1 of 2)

Device type	Description
ACLX devices	ACLX devices are device masking devices similar to VCM devices that are used for storage provisioning using Auto-Provisioning Groups on Enginuity 5874 and higher.
BCV devices	Specialized devices used to create a local copy of data contained in a standard array device, which can be used for backup, restore, decision support, and application testing. TimeFinder is a business continuance solution that allows you to use these special devices called BCV devices (Business Continuance Volume). Each BCV device has its own host address, and is configured as a stand-alone device.
DATA devices (datadev or TDAT)	(Enginuity 5876 or lower) DATA devices are used for Virtual Provisioning and are similar to SAVE devices, in that they are not visible to the host and must be contained in a pool before they can be used. Pools can only contain devices of the same emulation and protection type, however, the DATA devices can be different sizes.
Device Masking (VCM) devices	(Enginuity 5773 or lower) DMX devices that have been masked for visibility only to certain hosts. The device masking database (VCMDB) holds device masking records and typically resides on a 24 or 48 cylinder disk device. VCM devices are used for device masking for arrays running Enginuity 5773 or lower.
Diskless devices	(Enginuity 5876 or lower) The SRDF/Extended Distance Protection (EDP) streamlines a cascaded SRDF linkage out to the remote site with a direct (diskless) connection. In a cascaded SRDF/EDP environment, the cascaded R21 devices are designated as diskless devices. This provides replication between the source site and remote target site without requiring disks at the middle site. Because the diskless R21 device has no local disk space allocated to store the user data, it reduces the cost of having disk storage at the middle site.
DRV devices	(Enginuity 5876 or lower) Dynamic Reallocation Volume. A non-user-addressable device used by the Symmetrix Optimizer to temporarily hold user data while reorganization of the devices is being executed. Typically, it is used by the Optimizer in logical volume swapping operations.
Dynamic SRDF devices	Devices can be configured to be dynamic SRDF-capable devices. Dynamic SRDF functionality enables you to create, delete, and swap SRDF pairs while the array is in operation. Using dynamic SRDF technology, you can establish SRDF device pairs from non-SRDF devices, then synchronize and manage them in the same way as configured SRDF pairs. The dynamic SRDF configuration state of the array must be enabled via the configuration manager and the devices must be designated as dynamic SRDF-capable devices. For information about dynamic SRDF devices, refer to the *EMC Solutions Enabler SRDF Family CLI Product Guide*.
Gatekeeper devices	Gatekeeper devices are LUNs that act as the target of command requests to microcode-based functionality. For detailed information on gatekeeper management, refer to the *EMC Solutions Enabler Installation Guide*.
Metadevices	(Enginuity 5876 or lower) Allows individual devices to be concatenated to create larger devices. A metadevice consists of a metahead and its member devices. The *metahead* is the first device in the metadevice sequence and is responsible for receiving all incoming commands. It also identifies the entire metadevice. When an incoming command for the metahead is processed, the array determines which metadevice member should execute the command. Metahead devices can be added to a device group while a *metamember* cannot be added to a device group. This holds true for both standard and BCV devices.

Table 29 Device types - Enginuity 5876 and lower (page 2 of 2)

Device type	Description
SAVE devices	(Enginuity 5876 or lower) Special devices (not mapped to the host) that provide physical storage space for pre-update images or changed tracks during a virtual copy session of TF/Snap and SRDF/A DSE operations. SAVE devices are a predefined pool of storage devices and must be configured for this purpose. The SAVE device pool acts as a group for storing data in striped form. SAVE devices are assigned an array device number and can be unprotected, mirrored, or parity RAID. For more information, refer to the *EMC Solutions Enabler TimeFinder Family CLI Product Guide* and the *EMC Solutions Enabler SRDF Family CLI Product Guide.*
Standard devices	(Enginuity 5876 or lower) An array device configured for normal array operation under a desired protection method (such as RAID 1, RAID-5, and SRDF).
RDF devices	Devices configured as RDF1, RDF2, or RDF21 to support SRDF operations. SRDF is a business continuance solution that maintains a device-level mirror of array data on remotely attached arrays. These arrays also may be located in physically separate sites. RDF provides a recovery solution for component or site failures using remotely mirrored devices. SRDF reduces backup and recovery costs and significantly reduces recovery time after a disaster. For more information, refer to the *EMC Solutions Enabler SRDF Family CLI Product Guide.*
Thin devices (TDEV)	Thin devices for Virtual Provisioning are devices that do not have storage allocated to them when they are created. Thin devices can be created with an inflated capacity, because the actual storage space for the data written to the thin devices is on the DATA devices.To a host operating system, they look like regular devices with their configured capacity. The host treats them as regular devices and writes and reads from these devices like regular devices.
Virtual devices (VDEV)	(Enginuity 5876 or lower) A host-accessible device containing track-level location information (pointers), which indicates where the copy session data is located in the physical storage. Device copies use *virtual devices* to support TF/Snap operations. Virtual devices consume minimal physical disk storage, as they store only the address pointers to the data stored on the source device or a pool of SAVE devices. For more information, refer to the *EMC Solutions Enabler TimeFinder Family CLI Product Guide.*

Listing array device information

You can use two commands to list information about array devices:

- `sympd` — Lists array devices that are host-visible.
- `symdev` — Displays information about all array devices, host-visible or not.

The following `sympd list` command lists the physical device names with their corresponding array device names (`Sym`), director information, and device-specific information for all devices on array 064:

```
sympd list -SA 7E -sid 064

Symmetrix ID: 000197100064

        Device Name            Dir                  Device
---------------------------- ------- -------------------------------------
                                                                    Cap
Physical                Sym   SA :P   Config        Attribute    Sts   (MB)
---------------------------- ------- -------------------------------------
/dev/sdt                08028 07E:008 TDEV          N/Grp'd      RW    2063
/dev/sdu                08029 07E:008 TDEV          N/Grp'd      RW    2063
/dev/sdv                0802A 07E:008 TDEV          N/Grp'd      RW    2063
```

When using this command, you can specify a number of filter options to reduce the output of the command to the specific types of devices you are investigating. Limit the device output to a specific array by specifying the `-sid` option. List host-visible array devices that match a specific DA (`-da`), DX (`-dx`), interface (`-interface`), disk (`-disk`) or hyper-volume (`-hyper`) values. When specified along with the `-spindle` option, the output includes Spindle ID information. The `-vm` option displays valid virtual machine names on VMware ESX server environments.

You can also list the host-visible array devices that match a specific front-end director number (`-SA`) or front-end director port number (`-P #`). The output can be further limited to devices that are mapped to SCSI front-end directors (`-scsi`), Fibre front-end directors (`-fibre`), FICON front-end directors (`-ficon`), ESCON front-end directors (`-escon`), or Gig-E front-end directors (`-gige`).

While the `sympd` command only provides a limited amount of filter options, additional filter options on physical devices can be found by utilizing the `symdev` command with the `list pd` argument and associated options. Refer to "Filtering device data" on page 469 for more information.

Note: The `sympd show` and `sympd list -v` commands also include spindle ID information as part of both the back-end disk director Information section and the RAID group Information section.

The `sympd show` command's output reports spindle ID information as follows:

- For all but unprotected devices, the spindle ID will be reported as part of the RAID group Information section of the `sympd show` output.
- For unprotected devices, the spindle ID will be reported in both the RAID Group information section and the back-end disk director Information section.
- For non-RAID devices (VAULT devices), spindle IDs are reported in the back-end disk director Information section.

Note: For Enginuity versions lower than 5875, the specified/returned Spindle ID does not represent an actual Enginuity entity, but is a string representation of a unique hexadecimal number, which upon being returned may be used as an input parameter for `sympd show` commands.

The `symdev` command provides similar output as the `sympd` command, but includes all array devices and lists them by array device names.

The `symdev list` command also lists host-visible array devices that match specific DA (`-da`), DX (`-dx`), interface (`-interface`), disk (`-disk`) or hyper-volume (`-hyper`) values. Like `sympd`, when specified along with the `-spindle` option, the output includes Spindle ID information.

Note: The same Spindle ID reporting criteria described above for `sympd show` also applies to `symdev show`.

Viewing Recoverpoint Splitter devices (HYPERMAX OS 5977 only)

The `symdev list` command with the `-rp` option allows you to list array devices that are identified as RecoverPoint devices. Additional values for the RecoverPoint device type can be used to identify devices used as internal, production or replica devices `[-rp [Production | Replica | Internal]]`.

Viewing external spindle devices

Information about eDisks (external spindles) added using Solutions Enabler Federated Tiered Storage (FTS) can be viewed using the `symdev list` command with the `-external` option.

Federated Tiered Storage gives you the ability to use an external disk to attach external storage to VMAX Family arrays. Adding an eDisk to an array makes the eDisk's capacity available to the array as an external spindle.

Note: FTS is supported with Enginuity 5876 and HYPERMAX OS 5977.

With HYPERMAX OS 5977, FTS support is limited to virtual provisioning encapsulation of Data Domain eDisks for ProtectPoint. The `symdev list` command includes a new Encapsulated Device Flags field to indicate when an encapsulated device is a Data Domain device. In addition, a device status of Not Ready for a Data Domain device indicates that the device is Not Ready because it has been specified as the target of an image refresh.

The `symdev list` command uses the following syntax:

```
symdev [-sid SymmID] [-offline] [-v] [-resv | -pgr]
            [-wwn| -wwn_encapsulated [-detail]] [-all]

          list [ -FA <#|ALL> [-P <#>] |
            -SA <#|ALL> [-scsi] [-fibre] [-P <#>]]
        . . .
      [-technology <EFD | FC | SATA>]
            [-internal | -external | -encapsulated [-limited]]

          list pd [ -FA <#|ALL> [-P <#>] |
            -SA <#|ALL> [-scsi] [-fibre] [-P <#>]]
        . . .
      [-technology <EFD | FC | SATA>]
            [-internal | -external | -encapsulated [-limited]]

symdev [-sid SymmID] [-offline]

          list [-DA <#|ALL> | -DX <#|ALL>] -space [-cyl]
      [-spindle]

          list [-DA <#|ALL>] [-interface <#|ALL>]
            [-disk <#|ALL>] [-hyper <#|ALL>] [-firstport]
            [-spindle] [-internal]

         list [-DX <#|ALL>] [-hyper <#|ALL>]
            [-spindle] [-external]
```

The `symdev` options include:

- **`-DX`** — lists devices on a specific external director (DX) or all DX directors. The existing DA option only shows devices on an internal DA director.

- **-internal** — lists devices on internal spindles. TDEV, VDEV and DLDEV devices are not considered either internal or external and are not returned by this option. `-internal` can be abbreviated to `-int`.
- **-external** — lists devices on external spindles. TDEV, VDEV and DLDEV devices are not considered either internal or external and are not returned by this option. `-external` can be abbreviated to `-ext`.
- **-encapsulated** — lists encapsulated devices. `-encapsulated` can be abbreviated to `-enc`.
- **-limited** — lists devices that are geometry limited. These two options can also be used to report which TDEVs are considered encapsulated or geometry limited. Each of the -internal, -external and -encapsulated options can be abbreviated down to 3 characters.

Note: The symdisk list command also provides options for listing external spindle information. For more information, also refer to "Viewing external spindle information" on page 482.

RAID group information

The `symdev` command indicates, in verbose output, whether a rebuild or copy is in progress for a member of a device's RAID groups. The RAID Group Information section of `symdev show` output includes an additional parenthetical indicator under the Status column, which is displayed when a specific member is undergoing a rebuild (R), is the target of a member copy (C), or is in a failed state (F).

Example

The following example displays output for `symdev show`, but note that the output from `symdev list -v` also displays that same information:

```
symdev show 1234 -sid 123
   . . .
  Device Service State              : Normal
   . . .
  RAID Group Information
    {
   . . .
  RAID Group Service State      : Normal
   . . .
      Hyper Devices:
          {
          Device : 01234
              {
              ---------------------------------------------------------------
              Spindle   Disk    DA      Hyper        Member       Disk
                       DA :IT  Vol#  Num  Cap(MB)  Num Status Grp#  Cap(MB)
              ---------------------------------------------------------------
              528C     08A:D4    347   61       9    1 RW (R)    1   953870
              2D7      07C:C4    205   61       9    2 RW (C)    1   953870
              12C      07A:C6    134   61       9    3 RW        1   953870
              7E44     09C:D2    543   61       9    4 RW        1   953870
              }
```

Device states

A device can be in one of the following states:

- Ready
- Not ready
- Write disable
- Read/write enable

Changing the device state

To change the device state, set or reset the hold bit on a device, or relabel a device, use the following `symdev` command syntax:

```
symdev -sid SymmID [-devs <<SymDevStart>:<SymDevEnd> | <SymDevName>
              [,<<SymDevStart>:<SymDevEnd> | <SymDevName>>...]>]
[-noprompt] [-rp] [-celerra] [-star]

        rw_enable     [-SA <#|ALL>[-P <#>]]

        write_disable [-SA <#|ALL>[-P <#>]

        ready

        not_ready

        relabel

        hold

        unhold        [-symforce]
```

You can also specify a file with a list of devices that you want to change using the `-file FileName` option in the following `symdev` command:

```
symdev -sid SymmID -file FileName
       [-noprompt] [-celerra] [-star]

        rw_enable     [-SA <#|ALL> [-P <#>]

        write_disable [-SA <#|ALL> [-P <#>]

        ready

        not_ready

        relabel

        hold

        unhold        [-symforce]
```

Note: For VMAX arrays running HYPERMAX OS 5977 or higher, the `[-SA <#|ALL> [-P <#>]`options are not supported.

Viewing device service states

The `symdev list` and `symdev list pd` commands provide options that allow the returned data to be limited to a specific service state (`-service_state` *state type*).

There are three possible device service states:

- Normal
- Failed
- Degraded — indicates that one or more mirror positions of the protected device are not ready.

For example, the following command returns all devices on array 097 that are in a normal state:

```
symdev list -service_state normal -sid 097
```

The service states can also be listed to include all devices “not” in a particular service state.

For example, the following command returns all devices on array 097 that are not in the failed service state:

```
symdev list pd -service_state notfailed -sid 097
```

Device emulation

All host I/O transactions with an array of disk devices are managed by the Enginuity operating environment, which runs in the array I/O subsystem (channel directors and disk directors). Because each of the physical disks are indirectly seen as part of the I/O protocol, array devices are presented to the host with the following configuration or emulation attributes:

- Each device has **N** cylinders. The number is configurable (blocks ÷ 960).
- Each cylinder has **15** tracks (heads).
- Each device track in a fixed block architecture (FBA) has 64 blocks of 64K. (For non-FBA operating systems, the blocks are recognized without regard to the number of bytes.)

To return a list of configured devices by emulation type, enter:

```
symdev list -inventory

Symmetrix ID: 000192603139

    Device Config      FBA   CKD3390  CKD3380  AS400  CELERRA
  -----------------   -----  -------  -------  -----  -------
  Unprotected           256      N/A      N/A    N/A    N/A
  2-Way Mir            4040      N/A      N/A    N/A    N/A
  RAID-5                483      N/A      N/A    N/A    N/A
  RAID-6                515        8      N/A    N/A    N/A
  TDEV                  496      N/A      N/A    N/A    N/A
  DLDEV                2381      N/A      N/A    N/A    N/A
  RDF1+Mir                8      N/A      N/A    N/A    N/A
  RDF1+R-6               16      N/A      N/A    N/A    N/A
  RDF1+DLDEV           1275      N/A      N/A    N/A    N/A
  RDF2+Mir               40      N/A      N/A    N/A    N/A
  RDF2+R-5                8      N/A      N/A    N/A    N/A
  RDF2+R-6                8      N/A      N/A    N/A    N/A
  RDF21+R-5              16      N/A      N/A    N/A    N/A
```

```
RDF21+TDEV              16      N/A      N/A    N/A    N/A
BCV                     10      N/A      N/A    N/A    N/A
2-Way BCV Mir            5      N/A      N/A    N/A    N/A
BCV+R-5                  5      N/A      N/A    N/A    N/A
BCV+R-6                  5      N/A      N/A    N/A    N/A
2-Way DRV Mir           10      N/A      N/A    N/A    N/A
VDEV                   213      N/A      N/A    N/A    N/A
BCV+TDEV                32      N/A      N/A    N/A    N/A
```

This list can be limited to the devices in a specific array by specifying the array ID (`-sid`).

Showing details about a device

The `symdev show` command provides detailed information about a specific array device.

Note: Starting with HYPERMAX OS 5977, local RAID storage is managed using an internal algorithm. To support these changes and to present device details across all supported microcode levels, several changes have been made to the `symdev show` command. These changes present a common format for Mirror and RAID information regardless of RAID type and microcode version.

```
symdev show 8423 -sid 584
      . . .
    Mirror Configuration Information
        {
        Mirror Number                          : 1
        Mirror Type                            : RAID-6
        Mirror Status                          : Ready           (RW)
        }

    RAID Group Information
        {
        Mirror Number                          : 1
        RAID Type                              : RAID-6
        Device Position                        : Primary
        Protection Level                       : 6+2
        Disk Group Number                      : 1
        Disk Group Name                        : DISK_GROUP_001
        Engine Number                          : Multiple
        RAID Group Service State               : Normal
        Number of Failing Members              : 0
        Member Information:
            {
            Device : 8423
                {
                ------------------------------------------------------
                Spindle   Disk      Hyper      Member   Mir  Attr
                         DA :IT  Num Cap(MB) Num Status
                ------------------------------------------------------
                12FC     09C:D1   10     699   1 RW      1   N/A
                120C     07C:D1    9     699   2 RW      1   N/A
                111C     05C:D1    9     699   3 RW      1   N/A
                198C     07D:D0    9     699   4 RW      1   N/A
                1A7C     09D:D0    9     699   5 RW      1   N/A
                960      05B:C1   10     699   6 RW      1   N/A
                438      10A:C1   10     699   7 RW      1   N/A
                1914     06D:D1   10     699   8 RW      1   N/A
```

The following changes represent all Vault devices on Enginuity versions 5876 and lower. Vault devices are not supported for HYPERMAX OS 5977.

- The RAID Type in the RAID Group Information section is set to Unprotected for Vault devices, as only a single hyper is used.
- The Protection Level is set to 1 (same as an unprotected device).
- The RAID Group Service State is N/A as this is not available for Vault devices.
- The Number of Failing Members is N/A as this is not available for Vault devices.
- For arrays running Enginuity 5874 or higher, the Engine Number is set to the engine number where the Vault hyper resides.
- The Member/Num column in the Member Information section is set to 1, as Vault devices have a single hyper.
- The Member/Status column in the Member Information section is N/A as there is no comparable Mirror status for Vault devices to apply at the Member level.

For VDEV, TDEV, and similar devices, the RAID Group Information section is not displayed as there are no spindle-based members or hypers directly associated with these types of devices.

Showing clone state flags

The following CLI list and show commands report the Clone State Flags:

- `symdev show`
- `symdev list -v`
- `sympd list -v`
- `sympd show`
- `symdg show ld`
- `symdg list ld -v`
- `symcg show ld`

Note: Refer to the *EMC Solutions Enabler TimeFinder Family CLI Product Guide* for more information on the VP Snap feature and VSE.

HYPERMAX OS 5977supports TimeFinder SnapVX. A pair of new fields, Snapvx Source and Snapvx Target, indicate if the device is a TimeFinder SnapVX source or target. The Data Destaged field indicates if there is data de-staged to the device.

Enginuity 5876 supports TimeFinder VP Snap functionality. VP Snap leveragesT TimeFinder/Clone technology to create space-efficient snaps for thin devices by allowing multiple sessions to share capacity allocations within a thin pool. The clone Virtual Space Efficient (VSE) mode indicates a VP Snap session.

```
. . .
    Extent Based Clone       : None
    Snapvx Source            : True
    Snapvx Target            : False
    Data Destaged            : True
    DIF1 Flag                : False
    AS400_GK                 : False
    Host Cache Registered    : False
```

Showing disk geometry details

When the `-geometry` option is specified, the **Effective Device Geometry** field can have three possible values:

- **User Defined** —Indicates the user has defined geometry for this device.
- **Native** — Indicates the current device geometry is the same as the native geometry for the array.
- **Array wide emulation** — Indicates that the array-wide flag for FBA geometry emulation is set to enabled and the effective geometry shown in the output is derived from this setting.

Note: Even if the array-wide setting is turned on, the user can still define the geometry at the individual device level and this setting will take precedence over the array-wide setting. In addition, for non FBA devices, the CLI will indicate N/A with the `-v` and the `-geometry` options.

The following is an example of the device geometry output:

```
symdev show 0016 -geometry -sid 516

 Device Physical Name     : /dev/sdd
 Device Symmetrix Name    : 00516
 Device Serial ID         : 1600016000
 Symmetrix ID             : 000190300516

 Attached BCV Device      : N/A

 Attached VDEV TGT Device : N/A

 Vendor ID                : EMC
 Product ID               : SYMMETRIX
 Product Revision         : 5771
 Device WWN               : 60060480000190300516533030303136
 Device Emulation Type    : FBA
 Device Defined Label Type: N/A
 Device Defined Label     : N/A
 Device Sub System Id     : 0x0001

 Device Block Size        : 512

 Device Capacity
     {
     Cylinders            :       4400
     Tracks               :      66000
     512-byte Blocks      :    4224000
     MegaBytes            :       2063
     KiloBytes            :    2112000
     }

    Effective Device Geometry:    User Defined
     {
    Sectors/Track        :        128
    Tracks/Cylinder      :         15
    Cylinders            :       2000
    512-byte Blocks      :    3840000
    MegaBytes            :       1875
    KiloBytes            :    1920000

     }
```

```
Device Configuration     : RDF1            (Non-Exclusive Access)

Device is WORM Enabled   : No
Device is WORM Protected : No

SCSI-3 Persistent Reserve: Disabled

Dynamic Spare Invoked    : No

Dynamic RDF Capability   : None

STAR Mode                : No
STAR Recovery Capability : Sync_Tgt
STAR Recovery State      : Inactive

Device Service State     : Normal

Device Status            : Not Ready        (NR)
Device SA Status         : Ready            (RW)

Front Director Paths (2):
```

DATA and SAVE devices

Use the `symcfg` command to return details about all DATA devices, SAVE devices, and their pool memberships. The `-devs` and `-nonpooled` options can be used to filter the resulting output.

```
symcfg [-sid SymmID] [-offline] [-mb | -gb]

list [-savedevs [-fba] [-ckd3390] [-ckd3380] [-as400]
          [-nonpooled]
     [-devs [SymDevStart:SymDevEnd | SymDevName]
             [,SymDevStart:SymDevEnd | SymDevName...]]

        list [-datadev [-fba] [-nonpooled]
     [-devs [SymDevStart:SymDevEnd | SymDevName]
             [,SymDevStart:SymDevEnd | SymDevName...]]
```

When the `symcfg list` command is used to return pool information, the returned data will include pools that are currently associated with the group, or pools that have been disassociated from the group but still have the group's data.

When the `-all` option is specified, both enabled and disabled devices are included in the calculations of `Free Tracks` and `Full %`. Otherwise, only enabled devices are included. For example, without the `-all` option, the `Free Tracks` field includes free tracks from all enabled devices, and the `Full %` field is based on the enabled tracks. When the `-all` option is specified, the `Free Tracks` field includes free tracks from both enabled and disabled devices, and the `Full %` is based on the `Usable Tracks`.

For example, the following command returns a list of DATA devices for a specific array:

```
symcfg list -sid 237 -datadev
```

Thin device information

Use the `-tdev` option in the `symcfg list` command to return a list of thin devices in the system. You can filter the displayed list by emulation type or by a pool name to which the thin devices belong. The `-bound` or the `-unbound` option allows the listing of the thin devices to be filtered further. The `-unbound` option shows all the thin devices that are unbound, while the `-bound` option shows the bound thin devices.

```
symcfg [-sid SymmID] [-offline] [-mb | -gb]
list [-tdev
          [-devs SymDevStart:SymDevEnd | SymDevName
             [,SymDevStart:SymDevEnd | SymDevName...]]
          [-pool PoolName | [-fba] [-ckd3390]
             [-bound | -unbound]]
          [-sg SgName] [-detail | -tier]]
```

Listing thin devices

The following command lists thin devices on array 341. The columns `Allocated Tracks (%)` and `Written Tracks (%)` display the percentage of the entire thin device allocated and written.

```
symcfg list -tdev -sid 341

Symmetrix ID: 000194900341

Enabled Capacity (Tracks) :     785088
Bound   Capacity (Tracks) :     816420

                 S Y M M E T R I X   T H I N   D E V I C E S
-------------------------------------------------------------------------------
                                          Total          Total       Compressed
        Bound      Flgs      Total      Allocated       Written      Size/Ratio
Sym   Pool Name    EMPT     Tracks     Tracks (%)     Tracks (%)     Tracks (%)
----- ------------ ---- ---------- ---------- --- ---------- --- ---------- ---
0035B example2     F..B      33000      32796  99          0   0      32976   0
0035C test1        F..B      33000       1092   3       1075   3       1092   0
00A68 testing_341  F..B      33000      13428  41      13410  41      13428   0
00B1D xyz          F..B      14400         36   0          0   0         36   0
00B1E example2     F..B      48000         12   0          0   0         12   0
    . . .
Total                     ---------- ---------- --- ---------- --- ---------- ---
Tracks                     89557755     9560808  63    9518424  63    9560808   0

Legend:

 Flags:  (E)mulation : A = AS400, F = FBA, 8 = CKD3380, 9 = CKD3390
         (M)ultipool : X = multi-pool allocations, . = single pool allocation
         (P)ersistent Allocs : A = All, S = Some, . = None
         S(T)atus    : B = Bound, I = Binding, U = Unbinding, A = Allocating,
                       D = Deallocating, R = Reclaiming, C = Compressing,
                       N = Uncompressing, . = Unbound
```

Note: "Showing details about thin devices" on page 465 provides information on compressing and uncompressing thin devices.

The following example reports thin device tier allocations and displays both technology and disk locations.

```
symcfg list -tdev -sid 341 -tier

Symmetrix ID: 000194900341

               S Y M M E T R I X   T H I N   D E V I C E S
-------------------------------------------------------------------------------
                            Total            Tier
                Total     Allocated  Flag ------------------------------------
Sym   Emul      Tracks    Tracks (%)  LT  Protection   Name
----- ----- --------- --------- --- ---- ------------ -------------------------
01078 FBA       33000         0   0  IF  RAID-5(3+1)  FC_R5_VPTier
01079 FBA       33000         0   0  --  -            -
0107A FBA       66000       144   0  IS  RAID-6(6+2)  SATA_R6_VPTier
010EC FBA       33000       132   0  IF  RAID-6(14+2) FC_R6_VPTier
010EE FBA       33000         0   0  --  -            -
010F9 FBA        1500        20   0  X-  Unprotected  EXT_VPTIER
01128 FBA        8010        24   0  IF  RAID-6(14+2) Finance_VPTier
011A2 FBA       33000       216   1  --  -            [OutOfTier]

Total       --------- --------- ---
Tracks         240510       536   0

Legend:
Disk (L)ocation:
I = Internal, X = External, M = Mixed, - = N/A
(T)echnology:
S = SATA, F = Fibre Channel, E = Enterprise Flash Drive, M = Mixed, - = N/A
```

Listing thin devices in a pool

To display a list of thin devices in a pool, use the following sample syntax:

```
symcfg -sid 343 list -tdev -pool HR_THIN_R5
```

Showing thin BCV devices with dynamic SRDF capability

The following command lists device 15F9 as a thin BCV device enabled for either dynamic RDF1 or RDF2 capability:

```
symdev show 15F9 -sid 397

Device Physical Name     : Not Visible

Device Symmetrix Name    : 015F9
Device Serial ID         : N/A
Symmetrix ID             : 000194900397

Number of RAID Groups    : 0

Attached VDEV TGT Device : N/A

Vendor ID                : EMC
Product ID               : SYMMETRIX
Product Revision         : 5875
Device WWN               : 60000970000194900397533031354639
Device Emulation Type    : FBA
Device Defined Label Type: N/A
Device Defined Label     : N/A
Device Sub System Id     : 0x0016
Cache Partition Name     : DEFAULT_PARTITION
Bound Pool Name          : N/A
```

```
  . . .
Device Configuration     : BCV+TDEV

Device is WORM Enabled   : No
Device is WORM Protected : No

SCSI-3 Persistent Reserve: Enabled

Dynamic Spare Invoked    : No

Dynamic RDF Capability   : RDF1_OR_RDF2_Capable
```

Note: Bound Pool Name will be reported as N/A for an encapsulated device.

Listing data allocations across multiple pools

When you specify the `-detail` option, thin devices are listed with all the pools in which they have allocated tracks. The `Total Written Tracks` display for only bound pools and represent all tracks written to the thin device for any allocation in any pool, as shown in the following output:

```
symcfg -sid 397 -devs 1620:1630 list -tdev -detail

Symmetrix ID: 000195700397

Enabled Capacity (Tracks) :    2884812
Bound   Capacity (Tracks) :      20700

                 S Y M M E T R I X   T H I N   D E V I C E S
--------------------------------------------------------------------------------
                                    Pool         Pool           Total      Compressed
       Bound      Flgs       Total  Subs      Allocated        Written     Size/Ratio
Sym  Pool Name    ESPT      Tracks   (%)     Tracks (%)     Tracks (%)     Tracks (%)
---- ------------ ----- ---------- ----- ---------- --- ---------- --- ---------- ---
1623 testCust     F..B       14700    13       2400  16        744   0       1692  30
     test_pool    ----           -     -        252   2          -   -          0   0
1625 -            8...        1500     0          0   0          0   0          0   0
1626 -            8...        1500     0          0   0          0   0          0   0
1627 -            8...        1500     0          0   0          0   0          0   0
162E testCust     F..B        6000    10       4800  80       3600  60       1200  75

Total                    ---------- ----- ---------- --- ---------- --- ---------- ---
Tracks                       25200     1       7452  30       4344  17       2892  61

Legend:
 Flags:  (E)mulation : A = AS400, F = FBA, 8 = CKD3380, 9 = CKD3390
         (S)hared Tracks : S = Shared Tracks Present, . = No Shared Tracks
         (P)ersistent Allocs : A = All, S = Some, . = None
         S(T)atus    : B = Bound, I = Binding, U = Unbinding, A = Allocating,
                       D = Deallocating, R = Reclaiming, C = Compressing,
                       N = Uncompressing, . = Unbound
```

Monitoring space reclamation

Use the `symcfg list -tdev` command to monitor the progress of space reclamation. The following example output lists thin devices in the process of being reclaimed to free up unwritten allocated space (as a result of using the `symconfigure free` command). The `Reclaiming` status indicates the thin devices are in the process of being reclaimed. The allocated tracks and written tracks adjust accordingly as the space is reclaimed.

Showing details about thin devices

Use the `-detail` option with the `symcfg show` command to return details about a Virtual Provisioning pool. Any thin devices bound to a specified pool are listed along with thin devices that have allocated tracks within the pool but are not bound to that pool, as shown in `Other Thin Devices with Allocations in this Pool` in the following sample output.

If the `-all` option is added, the disabled devices in the pool are also shown.

The Pool State displays as Enabled, Disabled, or Balancing. If the Virtual Provisioning pool is enabled and the write balancing feature is turned on, the Pool State displays as Balancing. A thin device status of Reclaiming indicates the thin devices are in the process of being reclaimed as a result of using the `symconfigure free` command.

```
symcfg show -pool DG1_FBA_PL -thin -detail -sid 086

  . . .

Pool Name                        : DG1_FBA_PL
Pool Type                        : Thin
  . . .

Enabled Devices(208):
  {
   ------------------------------------------------------------
   Sym         Usable      Alloc       Free Full FLG Device
   Dev         Tracks     Tracks     Tracks  (%)  S  State
   ------------------------------------------------------------
   1FF30       274680         96     274680    3  .  Enabled
   1FF31       274680         96     274680    3  .  Enabled
     . . .
          ---------- ---------- ---------- ----
   Tracks     1056000      40332    1054464    3
  }

No Thin Devices Bound to Device Pool DG1_FBA_PL

Other Thin Devices with Allocations in this Pool (3):
  {
   ------------------------------------------------------------
                                        Pool        Compressed
           Bound           Total      Allocated     Size/Ratio
   Sym   Pool Name        Tracks     Tracks (%)     Tracks (%)
   ------------------------------------------------------------
   00003 -                 33000      32796  99          0   0
   00004 -                 49500       7536  15          0   0
     . . .
                     ---------- ---------- --- ---------- ---
   Tracks                82500      40332  48          0   0
  }

Legend:
  Enabled devices FLG:
    (S)hared Tracks : X = Shared Tracks , . = No Shared Tracks
  Bound Devices FLG:
    S(T)atus    : B = Bound, I = Binding, U = Unbinding, A = Allocating,
                  D = Deallocating, R = Reclaiming, C = Compressing,
                  N = Uncompressing, F = FreeingAll, . = Unbound,
```

Note: The **FLG S** flag field indicates whether there are shared allocations on each device as noted in the above legend.

Verifying thin and DATA device states

You can verify the state of thin or DATA devices using the following syntax:

```
symcfg -sid SymmID [-i Interval] [-c Count]
[-pool PoolName |
-g DgName |
-sg SgName |
-cg CgName |
-devs SymDevStart:SymDevEnd | SymDevName
[,SymDevStart:SymDevEnd | SymDevName...]

symcfg verify -tdev
[-bound | -binding | -allocating | -deallocating |
-unbound | -unbinding | -reclaiming]

symcfg verify -datadev
[-draining | -drainwait | -disabled | -enabled |
-deactivated | -nonpooled | -balancing]
```

More than one state may be tested. All devices must be in any of the requested states for the verification to succeed. Normally, the command executes a single poll and exits with status 0 (all devices in requested state) or nonzero status indicating which devices were not ready. A message, such as, “One or more of the specified devices are not in the requested state”, returns. If you require more than one poll to be executed, the default time between polls is 15 seconds.

The following options are available to report device states:

-dev — specifies the status of a single device returns.

-pool — all thin or DATA devices within a pool return, and must be in one of the requested states to pass verification.

-i — sets the polling rate. The default minimum interval value used for calculating the action is 15 seconds. A message displays at each poll.

-c — specifies the numbers of polls to be executed. The command will terminate at the end of the count or when all devices are in the correct state. If only the `-i` command is used, the command will loop infinitely, exiting only when all devices are in the correct state (or when manually interrupted).

Verifying device usage

For example, to continually check the status of a thin device (`5AC`) on array 1234 and exit the loop when it enters a bound state, enter:

```
symcfg -sid 1234 verify -tdev -dev 5ac -bound -I 5
```

To check once to see if all of the DATA devices in a pool (`devpool`) on array 1234 are disabled, enter:

```
symcfg -sid 1234 verify -datadev -pool devpool -disabled
```

Migrated devices

Both `symdev list` and `symdev show` provide information about devices that were changed or virtualized during a migration.

Changed devices

The `-identity_set` option of `symdev list` displays the devices whose effective identity changed after a migrate operation, as shown in the example output:

```
symdev -sid 207 -identity_set list

Symmetrix ID: 000192600207

        Device Name           Directors                  Device
--------------------------- ------------- -------------------------------------
                                                                          Cap
Sym  Physical               SA :P DA :IT  Config        Attribute    Sts   (MB)
--------------------------- ------------- -------------------------------------

0477 /dev/sddv              ***:* 06E:C1  2-Way Mir     N/Grp'd  (M) RW    2063
```

Virtualized devices

The `-identity` option of `symdev show` displays the effective identity information of devices that were virtualized during a migrate operation, as shown in the following example output:

```
symdev -sid 207 -identity show 0477

Device Physical Name     : Not Visible

Device Symmetrix Name    : 00477
Device Serial ID         : N/A
Symmetrix ID             : 000192600207

Number of RAID Groups    : 1

Attached BCV Device      : N/A

Attached VDEV TGT Device : N/A

Vendor ID                : EMC
Product ID               : SYMMETRIX
Product Revision         : 5825
Device WWN               : 60000970000192600207533030343737
Device Emulation Type    : FBA
Device Defined Label Type: N/A
Device Defined Label     : N/A
Device Sub System Id     : 0x0001
Cache Partition Name     : DEFAULT_PARTITION

Device Block Size        : 512

Device Capacity

    Cylinders            :        960
    Tracks               :      14400
    512-byte Blocks      :    1843200
    MegaBytes            :        900
    KiloBytes            :     921600

Effective Device Information
```

```
    Device WWN           : 60000970000192600306533030313732

    Front Director Paths (12): (See note below)

        ------------------------------------
         DIRECTOR   PORT             LUN
        ----------  ---- -------- ---------
        Type Num    Sts  VBUS TID SYMM Host
        ------------------------------------
        FA N/A  07E:3  RW   000  00  247  N/A
        FA N/A  07F:3  RW   000  00  247  N/A


    Geometry: User Defined
        {
        Sectors/Track    :         64
        Tracks/Cylinder  :         15
        Cylinders        :        200
        512-byte Blocks  :     192000
        MegaBytes        :         94
        KiloBytes        :      96000


Device Configuration     : Unprotected

Device is WORM Enabled   : No
Device is WORM Protected : No

SCSI-3 Persistent Reserve: Disabled

Dynamic Spare Invoked    : No

Dynamic RDF Capability   : RDF1_OR_RDF2_Capable

STAR Mode                : No
STAR Recovery Capability : None
STAR Recovery State      : NA

Device Service State     : Normal

Device Status            : Ready            (RW)
Device SA Status         : N/A              (N/A)
Host Access Mode         : Active
Device Tag(s)            : None


Front Director Paths (21):

    ----------------------------------------------------------------------
                            POWERPATH  DIRECTOR   PORT             LUN
                            --------- ----------  ---- -------- ---------
    PdevName                Type      Type Num    Sts  VBUS TID SYMM Host
    ----------------------------------------------------------------------
    Not Visible             N/A       FA   07E:1  RW   000  00  247  N/A
    Not Visible             N/A       FA   07F:1  RW   000  00  247  N/A

  . . .

Remote Copy Device Information

    Remote Copy Session Name               : N/A
    Session State                          : Created
    Pull                                   : True
    Offline Copy                           : False
    Differential Copy                      : Disabled
    Background Copy                        : Disabled
```

```
Incremental Copy                        : False
Donor Update                            : True
Federated Live Migration                : True
Starting Block                          :          0
Total Blocks                            :    192000
Percent Copied                          : N/A
Remote Devices [1]

  ----------------------------------------------------------------------
  Array:Dev              WWN                                Starting Block
  --------------------- --------------------------------- --------------
  000187990007:0012E     60000970000192600306533030313732              0
```

Pinned devices

A user-pinned device is a standard or thin device that cannot be moved by any FAST automated process but can be moved by the array Optimizer and the `symmigrate` operation. This functionality is available only on array arrays running Enginuity 5875 or 5876.

To pin the devices in group `prod1`, enter:

```
symdg -g prod1 pin
```

To unpin the devices in group `prod1`, enter:

```
symdg -g prod1 unpin
```

To list the pinned devices in array 341, enter:

```
symdev -sid 341 -pinned list

Symmetrix ID: 000194900341

        Device Name           Directors                  Device
--------------------------- ------------- -------------------------------------
                                                                          Cap
Sym  Physical               SA :P DA :IT  Config        Attribute    Sts   (MB)
--------------------------- ------------- -------------------------------------

 115 /dev/sdcq              08F:0 07A:D3  RAID-6        N/Grp'd      RW    2063
```

You can also view this information using the `symdev show` command.

Filtering device data

The `symdev list` and `symdev list pd` commands return a list of devices configured in one or more arrays connected to the controlling host. The `pd` qualifier returns only physical devices, and must appear after the list action on the command line, and is valid with all the `symdev list` options.

The `symdev list` command provides the following filter options to help you retrieve various types of information about your array devices:

```
symdev [-sid SymmID] [-offline] [-v] [-resv | -pgr]
            [-wwn | -wwn_encapsulated [-detail]] [-all]

         list [ -FA <#|ALL> [-p <#>] |
           -SA <#|ALL> [-scsi] [-fibre] [-p <#>]]
           [-devs SymDevStart:SymDevEnd | SymDevName
               [,SymDevStart:SymDevEnd | SymDevName...]]
           [-cap <#> [-captype <mb | cyl>]] [-N <#>]
           [-vcm | -aclx] [-held] [-gige]
```

```
[-ficon] [-escon] [-dldev] [-dd]
[-noport|-firstport|-multiport] [-bcv|-nobcv|-drv]
[-meta] [-nomember]
[-spare] [-dynamic] [-worm] [-vdev]
[-savedev [-nonpooled]] [-raids]
[-disk_group DskGrpNum | name:DskGrpName]
[-rg][-sec_raid] [-unprotected] [-sec_unprotected]
[-raid1] [-sec_raid1] [-bcv_emulation]
[-raid5 [-protection <3+1 | 7+1>]]
[-raid6 [-protection <6+2 | 14+2>]]
[-sec_raid5 [-sec_protection <3+1 | 7+1>]]
[-sec_raid6 [-sec_protection <6+2 | 14+2>]]
[-emulation fba|ckd|ckd3390|ckd3380|as400|celerra]
[-star_mode] [-star_sync_target] [-star_async_target]
[-reserved | -noreserved] [-cyl] [-geometry_set]
[-service_state [not]degraded | [not]failed |
                [not]normal]
[-tdev [-bound | -unbound]] [-datadev [-nonpooled]]
[-migr_tgt] [-pinned] [-host_passive] [-identity_set]
[-identity [-detail]] [-sg SgName]
[-rp [Production | Replica | Internal]]
[-technology <EFD | FC | SATA>]
[-host_cache] [-dif1] [-as400_gk] [-fast]
[-internal | -external | -encapsulated [-limited]]
[ -notrdf | [-rdfg RdfGrpNum [-rdfa]
[-R1] [-R2] [-R21] [-half_pair] [-dup_pair]]]
[-insg | -notinsg
[-host_id <compatible | native>]
```

The following sections provide additional details about some of the above options. For more information on these options, refer to the *EMC Solutions Enabler SYMCLI Command Reference Guide.*

Using the -all option

The `symdev` command has an `-all` option, which returns all configured devices.

Filtering by director

You can list the host-visible array devices that match a specific front-end director number (`-SA`) or front-end director port number (`-P #`). The output can be further limited to devices that are mapped to SCSI front-end directors (`-scsi`), or Fibre front-end directors (`-fibre`).

Filtering by director port mapping

In addition, you can limit the returned list of array devices to one of the following: devices that are mapped to more than one front-end director port (`-multiport`), devices that are not mapped to any front-end director port (`-noport`), or first port information for devices that have more than one port mapping schemas (`-firstport`). If none of these options are specified, then devices with all director-port relationships are returned.

Filtering by metadevice

You can list meta member devices by using the `-meta` option. The `-nomember` option prevents meta member devices from being displayed, but still displays meta head devices. These options are valid with any combination of `symdev` command options.

Filtering by SRDF devices

There are various `symdev` command options that can be used to filter SRDF devices. SRDF device filter options include: `-R1`, `-R2`, `-R21`, `-rdfg` *`RdfGrpNum`*, `-notrdf`, `-rdfa`, `-half_pair`, and `-dup_pair`. The `-R1`, `-R2`, and `-R21` device options can be used to list all of those types of SRDF devices respectively, and can be used together or independently to achieve the desired results. The `-rdfg` option only returns devices belonging to the the specified SRDF group number (*`RdfGrpNum`*). The `-notrdf` option returns all devices not configured as SRDF. The `-notrdf` option can also be used in conjunction with the `-R1`, `-R2`, and `-R21` device options to include non-RDF devices in the list along with those specified devices.

When used in conjunction with the `-dynamic` option, the `-notrdf` option can be used to return all dynamic capable devices that are not already SRDF devices. This allows you to quickly find non-SRDF devices that have dynamic SRDF capability.

Note: The `-R21` option is not currently supported in conjunction with the `-dynamic` option.

Use the `-rdfa` option to list all devices that are SRDF/Asynchronous capable. Use the `-half_pair` option to list all devices that are not partnered with an SRDF device. Existing half pair devices could have resulted from an SRDF/Star failover scenario, a half_deletepair operation, or a configuration change. Use the `-dup_pair` option to list all devices that are paired with the same SRDF type. Existing duplicate pair devices could have resulted from an SRDF/Star failover scenario or a configuration change.

Note: Refer to the *EMC Solutions Enabler SRDF Family CLI Product Guide* for detailed information on SRDF device types and failover configurations.

Filtering by FAST (Enginuity 5876 or lower)

The `symdev list` command with the `-fast` option is used to filter Fully-Automated Storage Tiering (FAST) controlled devices. This returns all devices from storage groups that are associated with either a FAST or FAST VP policy.

Solutions Enabler does not filter based on the policy it is associated with. For example, if you have a thick device under a storage group that is associated with a FAST VP policy, that device is returned as part of the result. Unbound and pinned devices are also returned as part of this filter.

Listing RAID device information

You can list RAID-S devices by RAID group using the `symdev list -raids` option (or `-raid` option, which performs the same operation).

Note: When viewing device output for RAID 5 devices, it is important to note that the total device capacity (`Device Capacity`) displayed is actually the total logical capacity. By contrast, the hyper capacity (`Hyper Num Cap`) displayed for each member is the physical capacity, which includes all device space—available and unavailable.

Filtering by technology type

By using the `-technology` option, you can filter on the drive technology (FC, EFD, and SATA) of the primary local back-end storage of the device. This is supported on arrays running Enginuity 5874 and higher.

Filtering by DA, interface, disk, or hypervolume

Host-visible array devices can be listed that match a specific DA (`-da`), DX (`-dx`), interface (`-interface`), disk (`-disk`), disk group number or name (`-disk_group`), or hypervolume (`-hyper`) values. The interface, disk, and hyper values default to ALL if the corresponding options are not specified.

Filtering by storage group

By using the `-sg` option, you can filter devices by a specified storage group.

Filtering by DIF1 attribute

The Data Integrity Field (DIF) is a setting on a device that is relevant to an Oracle environment and all hosts that support the DIF protocol. If the DIF1 attribute has been set on a device, output displays as `TRUE` or `FALSE` in the results of device list and show CLIs. The `symdev list` command with the `-dif1` option allows you to filter for devices that have the DIF1 attribute turned on.

DIF1 restrictions

The following restrictions apply while setting/resetting DIF1 attribute on devices:

- Only supported with Enginuity 5876, or higher.
- DIF1 attribute can only be set on FBA devices.
- DIF1 attribute can be set on both standard provisioned (thick) and virtually provisioned (thin) host accessible devices, setting the DIF1 attribute request on any internal devices will be rejected.
- Device must be unmapped while resetting the DIF1 attribute.
- Device can be mapped to only fiber front end directors (no iSCSI or FCoE) while setting the DIF1 attribute.
- If a device has DIF1 attribute already set then it can be mapped only to fiber front end directors (no iSCSI or FCoE).
- If creating a meta device, both the head and members need to have the DIF1 attribute in the same state (either all set or all reset).
- DIF1 attribute needs to be set before requesting reset. If the reset request is for a device range and if any one of the device has the DIF1 attribute not set, then the request will fail.
- DIF1 attribute can not be set on DATA and SAVE devices.
- Devices can only have the RDB_Checksum attribute or the DIF1 attribute set , but not both.
- The DIF1 flag can not be set on a device having active DCS.
- Devices can only have the ACLX attribute or the DIF1 attribute set, but not both.

- There is no relation between the DIF1 attribute and replication. Both source and target devices of any replication can have their own DIF1 setting.

Filtering by AS400_GK attribute

The AS400_GK attribute setting on an array device is required when a AS400 device is used in conjunction with IBM host control software. The attribute is also used in conjunction with the Celerra Network Attached Storage for Celerra gatekeeper devices.

If the AS400_GK attribute has been set on a device, output displays as `TRUE` or `FALSE` in the results of device list and show CLIs. The `symdev list` command with the `-as400_gk` option allows you to filter for devices that have the AS400_GK attribute turned on. Refer to "Setting device attributes" on page 241 for detailed information on setting the device attributes.

AS400_GK restrictions

The following restrictions apply while setting the AS400_GK attribute on devices:

- Only supported with Enginuity 5876 or higher.
- AS400_GK attribute can only be set on AS400 or Celerra FBA devices.
- The device cannot be configured as a SAVE, DATA, BCV, RDF or dynamic RDF device.
- If an AS400 or a Celerra FBA device is created using the `symconfigure create gatekeeper` command, then its AS400_GK attribute will be set to enable.
- The `symconfigure create gatekeeper`command is only supported for the following disk group provisioned emulations:
  - AS/400_M2107_A02
  - AS/400_M2107_A04
  - AS/400_M2107_A05
  - AS/400_M2107_A06
  - AS/400_M2107_A07
  - AS/400_M2107_A82
  - AS/400_M2107_A84
  - AS/400_M2107_A85
  - AS/400_M2107_A86
  - AS/400_M2107_A87
  - AS/400_M2107_050
  - AS/400_M2107_099
  - AS/400 D910_099
- The AS400_GK attribute can only be set on non meta devices.
- Setting the AS400_GK attribute is only supported on 2-48 cyl sized AS400 devices having the following disk group provisioned emulations:
  - AS/400_M2107_A02
  - AS/400_M2107_A04
  - AS/400_M2107_A05
  - AS/400_M2107_A06
  - AS/400_M2107_A07
  - AS/400_M2107_A82
  - AS/400_M2107_A84

- AS/400_M2107_A85
- AS/400_M2107_A86
- AS/400_M2107_A87
- AS/400_M2107_050
- AS/400_M2107_099
- AS/400 D910_099

Device output options

Specifying the `-cyl` option modifies the output list to include device capacity in cylinders rather than the default of megabytes (MB). The `-pd` option will only display the host visible devices (PDevs).

Releasing device external locks

The EMC Solutions Enabler uses *device external locks* in the array to lock pairs during replication operations (such as Open Replicator, TimeFinder, and SRDF operations).

To list a range of devices (0000 to 000A) that have a device external lock, enter:

```
symdev list -sid 870 -devs 0000:000A -lock
```

If you discover a lock from your host that has been held for well over two hours and you have confirmed no one is using the device resources, you can choose to release the lock. To release the device lock on a range of devices for array 870, enter:

```
symdev release -sid 870 -devs 0000:000A
```

⚠CAUTION

Use the release lock action only if you believe that the device lock was forgotten and there are NO other operations in progress to the specified devices (local or remote). Locks are typically of short duration (one second to an hour or so). However, it is critical to be able to recognize when a device lock being held by certain applications (such as an SRDF action) are allocated as long duration locks.

Aborting the session, if successful, will release all locks held by the session including device locks. The following examples show two instances of the `symdev release` command when the user attempts to release device locks held by the configuration server.

```
symdev release -sid 343 -devs 200:205

Within Symmetrix unit 000190300343, release lock for device 205 (y/[n]) ? y

Device lock is held by a configuration change session <654>. The session should be
   aborted to release the lock.

symdev release -sid 343 204

Within Symmetrix unit 000190300343, release lock for device 204 (y/[n]) ? y

Device lock is held by a migration session <654>. The session should be aborted to
   release the lock.
```

Disk-level data

The configuration database file maintains low-level configuration and status information for each disk on every array that is accessible from the host, as well as external spindles (eDisks). Using `symdisk list` and `symdisk show` commands, you can get a list of all the available disks. You can then choose a disk from this list and get its configuration and status information.

List and show disk information

For arrays running HYPERMAX OS 5977, the `symdisk list` command now accepts a new `-by_engine` qualifier to the `-dskgrp_summary` option. The command `symdisk list -dskgrp_summary -by_engine` displays a listing of attributes, counts and capacities relevant to Internal Disk Group/Engine Spindle Groups. The `symdisk list` and `symdisk show` commands also use new flags which indicate that disk capacity must be reported in megabytes, gigabytes, or terabytes:

```
symdisk [-sid <SymmID>] [-offline] [-cyl | -mb | -gb | -tb]
             [-disk_group <DskGrpNum | name:<DskGrpName> | ALL>
              [-all]] [-failed]

        list [-spindle [-internal]] [-isspare] [-nospares]
             [-v [-hypers] [-spare_info] [-gaps]]
             [-DA <# | ALL>] [-interface <# | ALL>]
             [-tid <# | ALL>]

        list [-spindle
               [-external [-detail] [-encapsulated [-free]]]]
             [-v [-hypers] [-gaps]]
             [-DX <# | ALL>]

     symdisk [-sid <SymmID>] -external -spindle -paths
             [-spid <SpindleID> |-DX <# | ALL> [-port <# | ALL>]]

         list -detail

         list [-offline]

     symdisk [-sid <SymmID>] [-offline] [-cyl | -mb | -gb | -tb]

        list -dskgrp_summary [-v]
             [-disk_group <DskGrpNum | name:<DskGrpName> | ALL>
              | -internal | -external]

        list -dskgrp_summary -by_engine [-v | -detail]
             [-disk_group <DskGrpNum | name:<DskGrpName> | ALL>
              | -internal | -external]

        show <DiskAddress> [-gaps_only]

     symdisk -sid <SymmID> [-offline] [-cyl | -mb | -gb | -tb]

        show -spid <SpindleID> [-gaps_only]

        show -wwn <ExternalWWN> [-gaps_only]
```

For arrays running Enginuity version 5773 through 5876, the `symdisk list` and `show` commands use the following syntax:

```
symdisk [-sid SymmID] [-offline] [-cyl]
             [-disk_group DskGrpNum | name:DskGrpName | ALL
              [-all]] [-failed]

        list [-spindle [-internal]] [-isspare] [-nospares]
             [-v [-hypers] [-spare_info] [-gaps]]
             [-DA <# | ALL>] [-interface <# | ALL>]
             [-tid <# | ALL>]

        list [-spindle [-external [-detail] [-encapsulated [-free]]]]
             [-v [-hypers] [-gaps]]
             [-DX <# | ALL>]

     symdisk [-sid SymmID] -external -spindle -paths
             [-spid SpindleID |-DX <# | ALL> [-port <# | ALL>]]
        list -detail
        list [-offline]

     symdisk [-sid SymmID] [-offline] [-cyl]
        list -dskgrp_summary [-v]
             [-disk_group DskGrpNum | name:DskGrpName | ALL]
              | -internal | -external]

        show DiskAddress [-gaps_only]

     symdisk -sid SymmID [-offline] [-cyl]
        show -spid SpindleID [-gaps_only]
        show -wwn ExternalWWN [-gaps_only]
```

Note: Do not specify the `-spare_info` and `-failed` options together in the `symdisk` command. This is an invalid combination.

The following example shows details about the specified disk:

```
symdisk show 07a:c0 -sid 432

Symmetrix ID                  : 000194900432
  Director                    : DF-7A
  Interface                   : C
  Target ID                   : 0
  Spindle ID                  : 2D0
    . . .
  Hypers (189):
  {
  #   Vol   Emulation        Dev   Type          Mir Mbr Sts Cap(MB)
  --- ----- ---------------- ----- ------------- --- --- --- --------
    1     1 VAULT_DEVICE     00000 Data           1  N/A N/A     5200
    2     2 FBA              001A9 RAID-5         1   1  WD     34179
   43    43 CKD-3390         000C0 RAID-1         1   1  NR       836
  189   189 FBA              00464 RAID-1         1   2  RW     31203
  }
```

When virtual devices (VDEVs) are part of your disk configuration, the capacity shown for these virtual hypers reflects only space to store track tables (pointers). Typically, a low number, such as 3 MB, is needed for these devices.

SAS devices

Serial Attached SCSI (SAS) disk drives in 300 GB, 450 GB and 600 GB capacity versions are supported. This includes the Cobra-D SAS 300, 450, and 600 GB disks drives (2.5" drive using 3.5" carrier and SAS/FC paddle card). The `symdisk show` and `symdisk list -v` CLI commands display the SAS disk technology type under the Technology label. The Form Factor label displays 2.5, 3.5, or N/A to note the Enterprise Flash Drive.

Note: For Disk Groups that contain internal spindles with different form factors (e.g. 2.5" and 3.5"), the Form Factor label displays as Mixed.

Disk gaps

Using the `-gaps` option with the `symdisk list` command lists any gaps found on the disk, and the `-spare_info` option tells you what disk the spare disk is substituting for, if it has been invoked.

The `symdisk show` command has a `-gaps_only` option that lists any gaps found on the disk without listing all of the hyper information.

Actual vs. rated disk capacity

The `Actual Disk Capacity` and the `Rated Disk Capacity` are calculated differently.

The `Actual Disk Capacity` reports on the physical disk capacities in MBs and GBs, where the MB is defined as (1024 x 1024) bytes and the GB is defined as 1024 MBs (or 1024 x 1024 x 1024 bytes).

The `Rated Disk Capacity` is based on the MB being defined as (1000 x 1000 bytes), the GB being defined as 1000 MBs (or 1000 x 1000 x 1000 bytes), and the TB defined as 1000 GBs. The `Rated Disk Capacity` is typically shown in sales and licensing documentation. This functionality is for arrays running Enginuity 5875 or higher.

Disk groups

When returning a list of disks, the `-by_diskgroup` option organizes the returned list of disks by disk group number. The `-disk_group` option allows you to limit the resulting list to disks belonging to a specified disk group. Disk groups can be specified by group number (*`DskGrpNum`*) or group name (`name:` *`DskGrpName`*). The `ALL` option returns disk information for all disk groups.

The `symdisk list -dskgrp_summary` command is used to display summary information for disk groups. Using the `-dskgrp_summary` option without the `-v` option displays disk group summary information in a table format. Using the `-dskgrp_summary` option with the `-v` option displays disk group information in expanded format (one field per line). Information regarding spare disks is not included in the `-dskgrp_summary` output.

Note: The `-dskgrp_summary` option is incompatible with the `-hypers`, `-spare_info`, `-gaps`, `-isspare` and `-failed` options.

Listing disk group summary for HYPERMAX OS 5977

For arrays running HYPERMAX OS 5977, the Hyper Size column is added, and this value represents the individual hyper size used for each Internal Disk Group/Engine Spindle Group. For external (FTS) Disk Groups or internal Disk Groups on Enginuity 5876 or lower versions the hyper size reports as ANY, since these groups do not use a consistent hyper size. The Free Capacity will be set to 0, since the Spindles in these Disk Groups are fully provisioned. The Actual Capacity column is removed.

For arrays running HYPERMAX OS 5977, the following example shows disk group summary information for all disk groups on array 584:

```
symdisk list -dskgrp_summary -sid 584

Symmetrix ID: 000197100584

      Disk Group                   Disk           Hyper         Capacity
----------------------- ----------------------- ------- -----------------------
                            Flgs Speed   Size    Size       Total       Free
Num  Name                 Cnt LT  (RPM)   (MB)    (MB)       (MB)        (MB)
----------------------- ----------------------- ------- -----------------------

   1 DISK_GROUP_001        67 IS   7200  1823565    3561   122178855           0
   2 DISK_GROUP_002        67 IF  15000  1823565    3561   122178855           0
 512 DISK_GROUP_512         1 X-    N/A      N/A     Any      204801      201114
 513 DISK_GROUP_513         1 X-    N/A      N/A     Any      204801      102400
                                                          ----------- -----------
Total                                                      244767312      303514
```

For the disk group summary verbose output:

- Actual Group Capacity and Summary Actual Group Capacity fields are removed.
- Max Hypers Per Disk and Hyper Size fields are added for each Disk Group.

```
symdisk list -dskgrp_summary -sid 584 -v

Disk Group                    : 1
  Disk Group Name             : DISK_GROUP_001
  Disks Selected              : 67
  Disk Location               : Internal
  Technology                  : SATA
  Speed (RPM)                 : 7200
  Form Factor                 : 3.5
  Disk Size (MB)              : 1823565
  Rated Disk Size (GB)        : 2000
  Total Group Capacity (MB)   : 122178855
  Free Group Capacity (MB)    : 23005664
  Max Hypers Per Disk         : 512
  Hyper Size (MB)             : 3561
.....
```

Listing disk groups with -by_ engine option

The `-by_engine` option can be used with the `symdisk list -dskgrp_summary` command to display a listing of attributes, counts and capacities specific to Internal Disk Group/Engine Spindle Groups.

```
symdisk list -dskgrp_summary -by_engine -sid 584

Symmetrix ID: 000197100584

         Disk             Hyper    Usable Capacity    Spare Coverage
------------------------ ------- ------------------- -----------------
               Flgs Speed  Size         Total          Total    Avail
Grp  Eng Cnt    LT  (RPM)  (MB)   Disk (%)    (MB)    Disk (%) Disk (%)
```

```
---- --- ---- ---- ----- ------- ---- --- ---------- ---- --- ---- ---
   1   1   50  IS   7200  113972   48  96   87530496    2   4    2 100
   1   2   50  IS   7200  113972   48  96   87530496    2   4    2 100
   1   3  100  IS   7200  113972   96  96  175060992    4   4    3  75
   2   4   50  IF  15000  113972   49  98  107589568    1   2    0   0
   2   5   25  IF  15000  113972   24  96   43765248    1   4    1 100
   2   6  100  IF  15000  113972   98  98  178708096    2   2    2 100
 512   7    1  X-    N/A     Any  N/A N/A     204800  N/A N/A  N/A N/A
         ----                     ---- --- ---------- ---- --- ---- ---
Total     376                      363  97  680389696   12   3   10  83

Legend:
  Disk (L)ocation:
    I = Internal, X = External, - = N/A
  (T)echnology:
    S = SATA, F = Fibre Channel, E = Enterprise Flash Drive, - = N/A
```

Abbreviations for the `-dskgrp_summary` option will be accepted from `-dskg` through the full option name. Abbreviations for the `-by_engine` option will be accepted from `-by_eng` through the full option name.

In addition to an Engine-based breakdown of values reported in the `-dskgrp_summary` display, the -dskgrp_summary -by_engine output includes the following:

- The Usable capacity (which does not include Spare capacity), reports in units of MB/GB/TB, based on -mb/-gb/-tb flag setting. The Usable capacity is also reported in units of disks and percentage (relative to the total capacity for the Disk Group/Engine Spindle Group) for internal disk groups for HYPERMAX OS 5977 and higher. The Disk and percent (%) columns are set to N/A for external (FTS) disk groups or for internal disk groups on Enginuity versions 5876 and lower.
- The Total and Available Spare Coverage values are reported in units of Disks and percentage for Internal Disk Group/Engine Spindle Groups. The Disk-based units correspond to the Disk Size and Max Hypers Per Disk reported in the `-dskgrp_summary -v` display. The Total Spare Coverage values include all Spare capacity, including unavailable capacity (i.e. failed disks), within each Internal Disk Group/Engine Spindle Group and the percentage is relative to the Usable capacity. The Available Spare Coverage values refer to the amount of Spare capacity that is currently available for sparing and this percent is relative to the Total Spare Coverage value. These columns report as N/A for external (FTS) disk groups or for internal disk groups on Enginuity versions 5876 and lower.

Listing disk groups with -by_ engine option including details

When the `-detail` flag is combined with the `-dskgrp_summary -by_engine` options, an expanded display is provided which adds the Disk Group Name.

`symdisk list -dskgrp_summary -by_engine -detail -sid 584`

```
Symmetrix ID: 000197100584

      Disk Group                 Disk             Hyper    Usable Capacity    Spare Coverage
----------------------- ------------------------ ------- ------------------- -----------------
                                     Flgs Speed  Size         Total          Total    Avail
Num  Name                      Eng Cnt LT  (RPM)  (MB)   Disk (%)    (MB)    Disk (%) Disk (%)
----------------------- ---- --- ---- ---- ----- ------- ---- --- ---------- ---- --- ---- ---
   1 DISK_GROUP_001          1   50  IS   7200  113972   48  96   87530496    2   4    2 100
   1 DISK_GROUP_001          2   50  IS   7200  113972   48  96   87530496    2   4    2 100
   1 DISK_GROUP_001          3  100  IS   7200  113972   96  96  175060992    4   4    3  75
   2 DISK_GROUP_002          4   50  IF  15000  113972   49  98  107589568    1   2    0   0
   2 DISK_GROUP_002          5   25  IF  15000  113972   24  96   43765248    1   4    1 100
```

```
   2 DISK_GROUP_002           6  100  IF  15000  113972   98  98  178708096    2   2    2 100
 512 DISK_GROUP_512           7    1  X-    N/A     Any  N/A N/A     204800  N/A N/A  N/A N/A
                                ----                      ---- --- ---------- ---- --- ---- ---
Total                    Total     376                   363  97  680389696   12   3   10  83

Legend:
  Disk (L)ocation:
    I = Internal, X = External, - = N/A
  (T)echnology:
    S = SATA, F = Fibre Channel, E = Enterprise Flash Drive, - = N/A
```

Spare physical disks

The `symdisk -isspare` command exclusively returns information about spare physical disks (spindles). If a spare disk is invoked against a failed disk, the `symdisk -v -spare_info` option can be used to return information about the failed disk that has been replaced. The `-v` option must be specified. Note that this display option is not applicable for permanent sparing.

Note: Dynamic sparing is only supported on arrays running Enginuity 5873 or lower. Permanent sparing is supported on arrays running Enginuity versions 5874 or higher.

The selection criteria allows you to return only data about the disks on a certain disk director (`-DA`), disk interface (`-interface`), or disk SCSI target ID (`-tid`). In addition, the `-isspare` option may be used to select only those disks that are configured as spare disks. The following example shows the truncated output, where the spare disk has been invoked against disk `16B:C0`:

```
symdisk list -v -spare_info
   ...
     Spare Disk                 : False

     Director                   : DF-16B
     Interface                  : C
     Target ID                  : A
     Disk Group Number          : 2

     Vendor ID                  : SEAGATE
     Product ID                 : SX3146807FC
     Product Revision           : CH146LF
     Serial ID                  : 3HY9CQVK

     Disk Blocks                : 0
     Block Size                 : 512
     Actual Disk Blocks         : 286749475
     Total Disk Capacity (MB)   : 0
     Free Disk Capacity (MB)    : 0
     Actual Disk Capacity (MB)  : 140014
     Hypers                     : 0

     Spare Disk                 : True

     Failed Director            : DF-15A
     Failed Interface           : C
     Failed Target ID           : 0
```

The `-gaps` option on the `list` command lists gaps found between hypers on the disk, as the hypers are listed. To see a short list of only the gap information, either do not specify the `-hypers` option on the `symdisk list` command, or specify the `-gaps_only` option on the `symdisk show` command.

Note: The gap sizes provided by this report are only an approximation. The report can be used as a general guide to the location and size of free space gaps, but it may not be accurate down to the last cylinder.

The `-nospares` option indicates that only internal disks (spindles) that are capable of being covered by a spare, but currently are not, should be displayed. This option is only supported on Enginuity version 5876 and higher. No disks (spindles) will be displayed if this option is issued for an array running a prior Enginuity version. In addition, this option is not applicable for external spindles (external spares do not exist) and no external spindles will be returned when this option is specified.

Spindle information

The `symdisk list` command with the `-spindle` option provides a disk listing that identifies each disk by Spindle ID and includes the director number for each. The `symdisk -spid` *`Spindle_ID`* option allows specifying physical disks by Spindle ID.

Note: For Enginuity versions lower than 5875, the specified/returned Spindle ID does not represent an actual Enginuity entity, but is a string representation of a unique hexadecimal number, which upon being returned may be used as an input parameter for `symdisk show` commands.

The following `symdisk list` command output shows an example of a disk listing using the `-spindle` option:

```
symdisk list -spindle -sid 306

Symmetrix ID                  : 000194900306
Disks Selected                : 64

                                                        Capacity(MB)
Spindle  Dir Vendor     Type       Hypr   Total       Free      Actual
-------- --- ---------- ---------- ---- ---------- ---------- --------
32E      08A SEAGATE    T146155      14     139028     120506     139028
16B35    08A SEAGATE    T146155      19     139028     118758     139028
1D83     08B SEAGATE    T146155      15     139028     123071     139028
22C34    08B SEAGATE    T146155      15     139028     124751     139028
 . . .
                                        ---------- ---------- ----------
Totals                                     7785556    6975177    7785556
```

The following `symdisk show` command shows sample output for a physical disk specified by Spindle ID (`-spid` *`Spindle_ID`*). Note that this is the same output format as `symdisk list -v` (verbose).

```
symdisk show -spid 1B935 -sid 306

Symmetrix ID                  : 000194900306
  Director                    : DF-8A
  Interface                   : C
  Target ID                   : 1
  Spindle ID                  : 1B935
  External WWN                : 60000970000195700233533030333132
  External Array ID           : 0001957000233
  External Device Name        : 031B
  Disk Group Number           : 001
  Disk Group Name             : DISK_GROUP_001
  Technology                  : FC
  Speed (RPM)                 : 15000
```

```
  Form Factor                    : N/A

  Vendor ID                      : EMC
  Product ID                     : Symmetrix
   . . .
```

Note: The `symdisk list` and `show` commands for spindles now identify remote Data Domain appliances when reporting on virtualized disks and LUNs on external arrays. If applicable, DataDomain displays in the Product ID field.

The following example shows `symdisk list` output for a specific disk group, with a request for the `-spindle` display, which includes director and Spindle ID information:

```
symdisk list -disk_group 004 -spindle -sid 306

Symmetrix ID                     : 000194900306
Disks Selected                   : 8
Disk Group                       : 4
Disk Group Name                  : DISK_GROUP_004
Technology                       : FC
Speed (RPM)                      : 15000
                                                        Capacity(MB)
Spindle  Dir Vendor     Type       Hypr   Total       Free      Actual
-------- --- ---------- ---------- ---- ---------- ---------- --------
32A      07A SEAGATE    HUC4515      52     418710     380208     418710
B7       08A SEAGATE    HUC4515      52     418710     380208     418710
5C9      07B SEAGATE    HUC4515      53     418710     379729     418710
1D01     08B SEAGATE    HUC4515      53     418710     364605     418710
 . . .
                                        ---------- ---------- ----------
Total                                      3349684    3009503    3349684
```

Note: The `sympd list` command with the `-spindle` option also provides spindle ID information. "Listing array device information" on page 452 provides more information.

Viewing external spindle information

Information on eDisks (external spindles) added using Federated Tiered Storage (FTS) can be viewed using the `symdisk show` command with the `-wwn` option. This command allows you to find the spindle ID of a newly added eDisk.

Federated Tiered Storage gives you the ability to use an external disk to attach external storage to a VMAX Family array. Adding an eDisk to a array makes the eDisk's capacity available to the VMAX array as an external spindle.

Note: FTS is supported with Enginuity 5876 and HYPERMAX OS 5977.

With HYPERMAX OS 5977, FTS support is limited to virtual provisioning encapsulation of Data Domain eDisks for ProtectPoint.

Refer to "Federated Tiered Storage" on page 301 for more information on FTS.

The output from the `symdisk` CLI command shows the Disk Location as *Internal* for internal disks or *External* for eDisks. The output also shows whether or not the eDisk is encapsulated. The Disk Service State represents the availability of the eDisk. For versions of Enginuity earlier than 5876, the Disk Location of the disks is shown as Internal. For external disks, the Technology shown for the disk displays as N/A. The number of external paths available to the external LUN to reach the eDisk is also displayed.

The Failed Disk State is reported as part of the Disk Service State, showing states of *Normal* or *Failed* for internal disks. For external disks the Disk Service State is shown as Failed if there are no network paths available to the eDisk (neither active nor passive), *Degraded* if there are paths from only one of its supporting DX directors to the eDisk (either active or passive), and Normal if there is at least one active and one passive network path available from both supporting DX directors to the eDisk.

CHAPTER 15
Statistics

This chapter describes how to use the SYMCLI `symstat` command to retrieve the performance-related statistics such as the number of I/O requests and throughput activities for the array devices.

- Overview 486
- REQUESTS statistics 487
- BACKEND statistics and PREFETCH activity 489
- Director statistics 490
- PORT statistics 491
- ISCSI statistics (GigE) 492
- MEMIO statistics 493
- Disk statistics 493
- CACHE statistics 494
- PATH statistics 496
- CYCLE statistics 497
- RDF statistics 498

Overview

The `symstat` command captures, in real-time, statistics information about an array. You can examine the performance of one or more devices, a device group, or any or all directors. For the capture of a sample period, `symstat` only displays devices or directors that had activity.

The `-c` argument defines the number of samples. The default for this argument is continuous sampling. If you do not specify this argument, but you specify an -i value, the command produces continuous statistical output, requiring a cancel (`Ctrl-C`) to stop the process. For the `symstat` command, a minimum default interval of 60 seconds is used for effective statistical sampling.

The `symstat` command performs the following:

- Queries array devices to capture raw performance counts and stores them in memory.
- Retrieves the performance counts for the array as a whole.
- Retrieves the performance counts for a director or director port.
- Retrieves the performance counts for one or more array devices.
- Retrieves the performance counts for one or more device groups, composite groups, or SRDF groups.
- Retrieves the performance counts for a selection of, or all, disks.
- Retrieves the timestamp of the performance count sample.
- Retrieves and displays replication session statistics for SRDF/A.
- Retrieves GigE iSCSI network statistics.

Note: Performance counters are not stored in the configuration database and only exist in the session context.

Reporting objects

The SYMCLI `symstat` command allows you to return valuable statistics on the how the following objects are performing in your environment:

- Devices — Statistical data can be returned on device throughput and I/O. Also, throughput activity to a specific set of devices within a device group can be obtained in addition to cache memory to disk activity for selected devices. Also, for arrays running Enginuity 5876 or lower, dynamic mirroring service policy (DMSP) statistics for the selected devices can be obtained.
- Directors — Throughput, back-end I/O, and port activity statistics can be obtained for a specific director or type of director.
- Disk — Reports back-end I/O requests and throughput for selected disks and tracks prefetch disk activity for selected back-end directors only.
- SRDF/A sessions — Monitor activity of SRDF/A sessions using the CYCLE, CACHE, and REQUESTS types in conjunction with the `reptype rdfa` flag of the `symstat` command.

Performance type options

The `symstat` command provides the performance display options listed in Table 30:

Table 30 Performance display options

Option	Description
REQUESTS	Reports I/O requests and throughput for selected devices, directors, or SRDF/A sessions. (This is the default type; if no type is specified, REQUESTS is used.)
BACKEND	Reports back-end I/O requests and throughput for selected devices.
PORT	Reports performance statistics for a director port.
ISCSI	Reports GigE network statistics.
CACHE	Reports cache activity for selected front-end or remote link directors, or SRDF/A sessions.
MEMIO	Reports cache memory to disk activity for selected devices.
PATH	Reports R-Copy path information for non-incremental sessions. Arrays that have all or some incremental sessions will report an error.
CYCLE	Reports cycle summary information for SRDF-A sessions.
DISK	Reports back-end I/O requests and throughput for selected disks.
PREFETCH	Reports track prefetch disk activity for selected back-end directors only.
DMSP	Reports dynamic mirroring service policy (DMSP) statistics for the selected devices. (Enginuity 5876 or lower)
RDF	Reports SRDF statistics from the perspective of RA groups, devices, or directors.

The following sections describe the `symstat` command options.

REQUESTS statistics

You can issue the symstat command with the REQUESTS type to display perf statistics such as I/O throughput for array device groups, individual devices, or SRDF sessions. REQUESTS is the default type for symstat operations, therefore, if no type is specified, REQUESTS performance information is returned.

Note: When using the `symstat` command against arrays running Enginuity 5874 or higher, some device and director output fields have been changed to support 64-bit counters. Refer to the"Setting compatibility" on page 31 for information on returning output as seen in prior Solutions Enabler versions.

Array statistics

I/O and throughput statistics can be returned for all devices on a specific array by specifying the Symmetrix ID in the command-line, specifying the interval and the number of samples (counts) for the statistical run. The following example returns performance statistics for array 345:

```
symstat -i 5  -type REQUESTS -sid 345
Symmetrix ID: 000194900345

         Sym                      IO/sec        KB/sec    % Hits  %Seq    Num WP
15:18:46 Dev   Physical         READ  WRITE   READ  WRITE RD  WRT RD  WRT Tracks

15:18:46 00383 Not Visible        0      0      0      0 N/A N/A N/A N/A      2
         0038F Not Visible        0      0      0      0 N/A N/A N/A N/A      3
                             ------ ------ ------ ------ --- --- --- --- ------
Total                             0      0      0      0 N/A N/A N/A N/A      5
```

The following statistics are returned:

1. Host timestamp of the sample.
2. Active device name and host physical device name (if the device is visible).
3. Read cache request rate (requests per second).
4. Write cache request rate (requests per second).
5. Read throughput rate (kilobytes per second).
6. Write throughput rate (kilobytes per second).
7. Read/write cache hit ratio (percentage of write hits).
8. Sequential read/write ratio (percentage of sequential reads).
9. Write-pending track count.

Device statistics

Using `symstat` with the REQUESTS type (default) specified, you can display performance statistics such as I/O requests using SAs (or other front-end adapters) and throughput on a device.

Select a logical device to examine I/O requests and throughput activity and choose the interval (`-i`) and number of samples (`-c`) for the statistical run.

For example, for sample intervals of `120` seconds and a sample count of `3` on logical device `DEV001` in device group `prod_r1`, enter:

```
symstat -i 120 -c 3 -g prod_r1 -ld DEV001
```

The fields returned are identical to those returned in "REQUESTS statistics" on page 487, except that the statistics are limited to the one logical device specified.

SRDF/A session statistics

Using `symstat` with the REQUESTS type specified, you can display a table of host writes and SRDF throughput information for SRDF/A sessions in arrays by specifying the `-repType rdfa` option.

When requesting SRDF/A session statistics for a specified array, the output differs depending upon the flag arguments specified:

- If the `-sid` flag is used with no further qualifiers (`[-g ALL | -rdfg <#|ALL> | -cg CgName|ALL]`), then the output display lists SRDF/A sessions that have a status of `active`.
- If the `-sid` flag is used with qualifier `-g all`, then the output display list all device groups, regardless of status. If the `-sid` flag is used with the qualifier `-rdfg ALL`, then the output display lists all SRDF groups, regardless of status. If the `-sid` flag is used with the qualifier `-cg ALL`, then the output display lists all composite groups, regardless of status.
- If the `-g DgName` device group qualifier is used (whereby the `-sid` option is not specified), then the output display only lists details for the specified device group.
- If the `-sid` flag is used with the qualifier `-rdfg #`, then the output display lists just the specified SRDF group.
- If the `-sid` flag is used with the qualifier `-cg CgName`, then the output display lists just the specified composite group.
- If the `-v` option is specified, the returned statistics are expanded and grouped by device group (SRDF/A session) for all sessions.

R-Copy session statistics

Using `symstat` with the REQUESTS type specified, you can display a table of host writes and SRDF throughput information for Open Replicator R-Copy sessions in arrays by specifying the `-repType rcopy` option.

You can refine the data returned for R-Copy sessions by specifying an optional session name as shown in the following syntax:

```
symstat [-i Interval] [-c Count] [-type REQUESTS] -sid SymmID
   -RepType rcopy [-name SessionName]
```

To return R-Copy session statistics for array 0003, enter:

```
symstat -sid 0003 -reptype rcopy -i 60 -c 3
```

Mainframe (Ficon) statistics

Using `symstat` with the REQUESTS type specified, you can display a table of mainframe Ficon throughput information in arrays by specifying the `-EF` option. You can either specify a specific port number or all port numbers.

BACKEND statistics and PREFETCH activity

The `symstat` command can return statistics for I/O requests on back-end directors, the throughput on the corresponding devices, and prefetch disk activity on back-end directors.

Returning BACKEND statistics

Using `symstat` with the BACKEND type specified, you can display performance statistics such as I/O requests using a DA (back-end director) and throughput on a device.

From the device list shown in the device group statistical run, select the logical device for which you want to examine I/O requests and throughput activity. Choose the interval and number of samples (count) for the statistical run.

For example, for sample intervals of `60` seconds and a sample count of `3` on logical device `DEV001` in device group `prod_r1`, enter:

```
symstat -type BACKEND -i 60 -c 3 -g prod_r1 -ld DEV001
```

The display would be similar to the front-end REQUESTS display showing read/write I/O, prefetched tracks, and unused track activity.

Prefetch activity

From the list, select the array that you want to examine for prefetch disk activity on the back-end directors (DAs). Choose the interval and number of samples (counts) for the statistical run.

For example, for sample intervals of `60` seconds and a sample count of `4` for prefetch activity on array `010050010333`, enter:

```
symstat -i 60 -c 4 -sid 333 -da all -type prefetch
```

Solutions Enabler with Enginuity 5876 or HYPERMAX OS 5977 provides the `symstat` command with the `-DX` option to allow prefetch disk activity on external directors (DXs). Choose the interval and number of samples (counts) for the statistical run.

For example, for sample intervals of 60 seconds and a sample count of 4 for prefetch activity on array 010050010333, enter:

```
symstat -i 60 -c 4 -sid 333 -dx all -type prefetch
```

Director statistics

Using `symstat`, you can display the I/O requests and throughput activity performance statistics on any or all directors. In addition, you can return director statistics for R-Copy sessions. To return a list of directors available on all visible arrays, use `symcfg list -dir all`.

Note: You can run statistics for a FICON directors. The `-dir` option of `symstat` accepts a FICON director as an argument for arrays running Enginuity 5874 or higher.

Director I/O requests

From the list, select the array that you want to examine I/O requests and throughput activity on the directors. Choose the interval and number of samples (counts) for the statistical run.

For example, to output statistics for all directors on array 150, enter:

```
symstat -type REQUESTS -sid 150 -dir ALL -i 60 -c 100
```

The following statistics are returned:

- Host timestamp of sample.
- Active director identifier and number.
- I/O rates (operations per second) for:
  - Host to Front-end directors
  - Disk to Back-end directors
  - Local to Remote link directors
- Read cache request rate (requests per second).
- Write cache request rate (requests per second).
- Total read/write cache request rate (requests per second).
- Cache hit ratio (percentage of cache read and write hits).

R-Copy session statistics

Using `symstat` with the REQUESTS type (default) specified, you can display a table of directors on the specified array and SRDF throughput information for Open Replicator R-Copy sessions in arrays by specifying the `-repType rcopy` option.

You can refine the data returned for R-Copy sessions by specifying an optional session name as shown in the following syntax:

```
symstat [-i Interval] [-c Count] [-type REQUESTS] -sid SymmID
   -RepType rcopy [-name SessionName]
```

To return R-Copy session statistics for array 097, enter:

```
symstat -sid 097 -reptype rcopy -dir 7a -i 60
```

The following statistics are returned:

- Host timestamp of sample.
- Active director identifier and number.
- R-Copy director throughput in MB/sec.
- Number, or the count, of cache read requests per second.
- Number, or the count, of write requests per second.
- Total number, or count, of combined read and write requests per second.
- Cache hit ratio (percentage of cache read and write hits).

PORT statistics

Using `symstat` with the PORT type specified, you can display performance statistics such as I/O requests and throughput of director ports for a specific array.

How to return director port statistics

Choose the array for which you want to examine I/O requests and throughput activity on the directors. Choose the interval and number of samples (counts) for the statistical run. The `symstat -type PORT` syntax allows you to return statistics for one or all directors on one or all arrays. In addition, you can optionally specify one or all port numbers. Use the following syntax for director port statistics:

```
symstat [-i Interval] [-c Count] -type PORT [-sid SymmID]
   -dir <#|ALL> [-port <#|ALL>]
```

For example, for sample intervals of `60` seconds, and a sample count of `3` (for I/O requests and throughput) on array `0256`, enter:

```
symstat -i 60 -c 5 -type PORT -sid 0256 -dir ALL
```

R-Copy session director port statistics

Using `symstat` with the PORT type specified and the `-reptype` set to `rcopy`, you can display a table of director ports on the specified array and session throughput information for Open Replicator R-Copy sessions.

The following statistics are returned:

- Host timestamp of sample.
- Active director and port number.
- R-Copy session throughput in MB/sec.
- The maximum percentage of that director/port's bandwidth that is allotted to R-Copy operations.
- Number of remote devices. The number of remote devices is only available when all R-Copy sessions are non-incremental.

ISCSI statistics (GigE)

Using `symstat` with the iSCSI type specified, you can display GigE performance statistics, such as I/O requests and throughput of director ports for a specific array. Statistics can be returned for GigE directors configured as SRDF directors (`-RE`) and for GigE directors (`-DIR`) configured as SCSI directors.

- To return TCP connection-based statistics, specify the `-tcp_conn` option and the SCSI director number (ALL can be specified as a value). For example, enter:

  ```
  symstat -type ISCSI -i 60 -c 5 -sid 237 -tcp_conn -DIR 20
  ```

- To return network interface-wide statistics, specify the `-gige_dir` option and the SCSI director number (ALL can be specified as a value). For example, enter:

  ```
  symstat -type ISCSI -i 60 -c 5 -sid 237 -gige_dir -DIR 20
  ```

- To return SRDF speedlimit statistics, specify the `-rdf_spdlmt` option and the SRDF director number (ALL can be specified as a value). For example, enter:

  ```
  symstat -type ISCSI -i 60 -c 5 -sid 237 -rdf_spdlmt -RE 24
  ```

- To return SRDF network compression statistics, specify the `-rdf_nw_comp` option, SRDF director number (ALL can be specified as a value), and the IP address of the remote array. For example, enter:

```
symstat -type ISCSI -i 60 -c 5 -sid 237 -rdf_nw_comp -RE 24 -ip 192.168.100.150
```

- To return SRDF speedlimit group statistics, specify the `-spdlmt_grp` option and the SRDF director number (ALL can be specified as a value). For example, enter:

```
symstat -type ISCSI -i 60 -c 5 -sid 237 -spdlmt_grp -RE 24
```

MEMIO statistics

Using `symstat` with `MEMIO` type, you can display performance statistics such as cache memory to disk I/O on a device group. However, you may need to first list the current device groups (use `symdg list)`. From the device group list, select the device group for which you want to examine the I/O or cache memory to disk activity. Choose the interval and number of samples (counts) for the statistical run.

The following display is an example output of cache to disk activity for the devices of array 150:

```
symstat -i 5 -g Prod -ld DEV001 0383 -type MEMIO -sid 345
Symmetrix ID: 000194900345

                     Sym                        Tracks/sec    % Dev
15:39:44 Logical     Dev         WP Tracks   Prefchd Destgd  WPmax
15:39:44 DEV001      00383               1          0      0      0
```

The following statistics are returned:

- Host timestamp of sample and array device name.
- Write pending track count not yet destaged to disk.
- Track prefetch rate (tracks per second).
- Track destage rate (tracks per second).
- Write-pending device limit percentage (percentage of device parameter).

By default, BCV devices are not included in these results. Optionally, you can choose to return statistics for only BCV devices in the device group by specifying the `-bcv` option. Alternatively, you can choose to return statistics for both standard and BCV devices in the device group by specifying the `-all` option.

Disk statistics

Using `symstat` with the DISK type specified, you can display performance statistics such as I/O requests and throughput on a physical disk, at specified time intervals for the duration of a specified count.

Specifying a disk

Select the physical DA director number, interface ID, and disk SCSI ID interface on a specified array for which you want to examine I/O requests and throughput activity. To retrieve the statistics of all DA directors, interfaces, or disk SCSI IDs, use the value ALL. For example, for sample intervals of `60` seconds and a sample count of `3` on disk `02A:C5`, enter:

```
symstat -type disk -i 60 -c 3 -sid 577 -disk 2a,C,5
```

If you want to retrieve statistics at the same interval and sample count, but for all SCSI IDs on DA 02A, interface C, enter:

```
symstat -type disk -i 60 -c 3 -sid 577 -disk 2a,C,ALL
```

The display provides the disk ID, I/O read/writes, and kilobytes per read and write for each disk.

Returning spindle statistics

For arrays running Enginuity 5876 and higher provides the `symstat` command with the `-spindle` option to allow the identification of disks by Director and Spindle ID for internal or external (DX director) disks. Also the `-spid` *`SpindleID`* option allows statistics to be requested for a disk through the specification of a Spindle ID or for all disks. The `symstat` command uses the following syntax:

```
symstat -i Interval [-c Count]
        -type DISK -sid SymmID [-spindle [-internal|-external]]
        [[-disk <da|ALL>[,<int|ALL>[,<id|ALL>]]] |
         [-spid SpindleID|ALL]] [-DX <#|ALL>]]
```

For example, to request an output format which identifies disks by director and Spindle ID for array 109, enter:

```
symstat -type DISK -sid 109 -spindle -i 60
```

Note: For Enginuity versions lower than 5875, the specified/returned Spindle ID does not represent an actual Enginuity entity, but is a string representation of a unique hexadecimal number.

CACHE statistics

You can return data about the cache activity on the front-end directors, LRU, or SRDF/A sessions by specifying the CACHE type with the symstat command.

Note: LRU statistics are not available for arrays running HYPERMAX OS 5977 or higher.

Front-end director statistics

You can return statistics for front-end directors by specifying options for SA, EA, FA, FE, CA, RA, director number, or all. In addition you can choose the interval and number of samples (counts) for the statistical run. For example, to return cache activity for all SAs on array `010050010333`, for sample intervals of `60` seconds and a sample count of `4`, enter:

```
symstat -i 60 -c 4 -sid 097 -sa all -type cache
```

SRDF/A session statistics

Use the `symstat` command to return a table of cache usage for device groups (SRDF/A sessions) in the specified array that have active status or have nonzero cache usage by specifying the `-repType rdfa` option. SRDF/A cache statistic data can be returned for all or a specific device group or RA group number.

When you request SRDF/A session statistics for a specified array, the output differs depending on the flag arguments specified:

- If the `-sid` flag is used with no further qualifiers (`[-g ALL | -rdfg <#|ALL> | -cg <CgName|ALL>]`), the output displays SRDF/A sessions that have a status of `active`.
- If the `-sid` flag is used with qualifier `-g all`, the output displays all device groups, regardless of status. If the `-sid` flag is used with the qualifier `-rdfg ALL`, the output displays all SRDF groups, regardless of status. If the `-sid` flag is used with the qualifier `-cg ALL`, the output displays all composite groups, regardless of status.
- If the `-g DgName` device group qualifier is used (whereby the `-sid` option is not specified), then the output displays only details for the specified device group.
- If the `-sid` flag is used with the qualifier `-rdfg #`, then the output displays just the specified SRDF group.
- If the `-sid` flag is used with the qualifier `-cg CgName`, then the output displays just the specified composite group.
- If the `-v` option is specified, the returned statistics are expanded and grouped by device group (SRDF/A session) for all sessions.

Example: R1 SRDF/A session statistics for active groups

For example to return SRDF/A session statistics for all active device groups on array `0011`, enter:

```
symstat -type cache -i 60 -c 3 -sid 0011 -repType rdfa
```

The following SRDF/A session statistics are returned:

- **System Write Pending Limit** — The number of cache slots available in the array, which are used to hold writes awaiting destaging locally or transfer in an SRDF/A cycle. Typically, this limit is approximately 80 percent of the total cache available, however, in some case it is less.
- **Cache Slots Available for all SRDF/A sessions** — The upper limit of cache slots available for all SRDF/A sessions. This value typically is determined by the SRDF/A Max Cache Usage parameter, which allows the user to specify what percentage of the System Write Pending Limit slot count should be used for SRDF/A sessions. If the Max Host Throttle parameter is 0, then Cache slots available for all SRDF/A sessions is a percentage of System Write Pending Limit as specified by SRDF/A Max Cache Usage setting. The default value of this limit is approximately 94 percent of the System Write Pending limit. If the Max Host Throttle parameter is not 0, then the number of Cache Slots Available for all SRDF/A sessions will be equal to the System Write Pending limit.
- **Total Local Write Pending Count** — The sum of local write pending tracks for all RDF devices. Note this is for RDF devices ONLY.

- **Total System Write Pending Count** — This count represents all write pending tracks in the system, which in most cases sum of write pending tracks for all of the devices. There are cases when the same write pending track belongs to more than one device. This includes all devices that are RDF.
- **Total System SRDF/A DSE Used Tracks —** Identifies the SRDF/A DSE threshold and used track counts.
- **Total System Cache Slots in Use** — Cache slots in use is the total number of cache slots in the active and inactive cycle for that SRDF/A session. These are tracks that are used by the SRDF/A session but are not directly related to write pending tracks. Tracks can be write pending, slots in memory, or in DSE pool. Therefore, using this number in combination with any write pending count does not produce any meaningful value.

  Note: The sum of the Cache Slots in Use and the Local Write Pending Count does not equal the Total System Write Pending Count. This is because Cache Slots in Use and Local Write Pending Count include every cache slot in those states, while the Total System Write Pending Count includes slots with either of those states, but not both. So a cache slot with both a local write pending and an SRDF/A write pending would increase the individual counts of each, but only increment the total count by one.

- **Total System Cache Slots Shared** — A count of total system cache slots that measures the amount of data shared.
- **Total System Cache Full Percentage** — The percentage of the available SRDF/A cache in use for the device group.

PATH statistics

You can report R-Copy path information for non-incremental sessions. Arrays that have all or some incremental sessions will report an error.

Note: These statistics are only available when all R-Copy sessions are non-incremental.

The data can either be presented by session, where the data is organized by session name (`-by_session` option), or by port, where the data will be organized by director port (`-by_port` option).

The following example shows the R-Copy PATH statistics for array 003 organized by director port:

```
symstat -sid 0003 -reptype rcopy -type path -by_port -i 60 -c 3

Time       Director   RCopy    Port   RCopy   Control      Tracks       Session
Stamp        Port    Ceiling  KB/sec  KB/sec  Device   Total Remaining   Name

15:44:44  FA-14C:1  30            -       -  00644    264960       0  hotpull
                                             00645    264960       0  hotpull
                                             00646    264960       0  hotpull
                                             00647    264960       0  hotpull
```

CYCLE statistics

The CYCLE type statistics return a table of cycle information for device groups (SRDF/A sessions) in arrays that have active status or have nonzero cache usage.

When you request SRDF/A session statistics for a specified array, the output differs depending on the flag arguments specified:

- If the `-sid` flag is used with no further qualifiers (`-g all`, `-rdfg # | ALL`), the output displays SRDF/A sessions that have a status of `active`.
- If the `-sid` flag is used with qualifier `-g all`, the output displays all device groups, regardless of status. If the `-sid` flag is used with the qualifier `-rdfg all`, the output displays all SRDF groups, regardless of status.
- If the `-sid` flag is used with the qualifier `-rdfg #`, the output displays just the specified SRDF group.
- If the `-g` *`DgName`* device group qualifier is used (without specifying the `-sid`), the output displays only details for the specified device group.
- If the `-v` option is specified, the returned statistics are expanded and grouped by device group (SRDF/A session) for all sessions.

Example: All device groups

For example, to return SRDF/A session cycle statistics of active sessions for array `0011`, enter:

```
symstat -type cycle -i 60 -c 3 -sid 0011 -RepType rdfa -g all
```

The following statistics are returned for one count:

```
SRDF/A Session Cycle Summary Information
Symmetrix Id  : 000187400011
Timestamp     : 14:40:16


Session    Cycle Time  (sec)        Cycle Size
                       ----------- ------------------- -----------------------
                  RA   FLGS Active                Last
Device Group Name Grp  TAS  Cycle# Min Avg Last Switch Active  Inact   Shared
----------------- ---  ---- ------ --- --- ---- ------ ------- ------- -------
RaGrpNum_111      111  2YA     853   -  15   15      1       0       0       0
RaGrpNum_112      111  2YA     753   -  15   15      1       0       0       0
RaGrpNum_121      111  2YA     853   -  15   15      2       0       0       0
RaGrpNum_122      111  2YA     853   -  15   15      1       0       0       0
RaGrpNum_123      111  2YA     855   -  15   15      1       0       0       0
RaGrpNum_124      111  2YA     856   -  15   15      1       0       0       0

Legend for Session Flags:

  T(ype)          : 1 = RDF1, 2 = RDF2
  A(SYNC)         : Y = Yes, N = No
  S(tatus)        : A = Active, I = Inactive, - = N/A
```

Note: The table headings Active and Inactive are used to represent the Captured and transmitted data on the R1 side and Receive and Apply on the R2 side of the operation.

The following statistics are returned:

- Device Group Name
- RA Group number
- Session Flags
  - Type of SRDF group (RDF1 or RDf2); inactive sessions do not have an assigned type.
  - Asynchronous (Yes or No).
  - Status (Active or Inactive).
- Active cycle number identifier for the given SRDF/A session. MSC cycle numbers are not displayed.
- Cycle Time
  - Min — The setting for the minimum number of seconds for a specific cycle.

    Note: The Min Cycle Time value is not shown on the R2 side.

  - Avg — A measure of the average time of all cycles in the session in seconds.
  - Last — A measure of the last cycle in seconds.
  - Last Switch — A measure of time in seconds since the last switch from active to inactive or vice versa.
- Cycle Size
  - Active — A count of cache slots that measures the amount of data captured (on the R1 side) or received (on the R2 side).
  - Inactive — A count of cache slots that measures the amount of data transmitted (on the R1 side) or applied (on the R2 side).
  - Shared — A count of cache slots that measures the amount of data shared.

RDF statistics

The RDF type statistics extend the statistical information provided by the RDF director to external applications providing greater visibility into the performance and behavior of arrays in the field. These statistics can be used to monitor on-going activity and to analyze problematic behavior.

Note: RDF directors are supported on VMAX 10K arrays running Enginuity 5876. A `symstat` command specifying a RDF director using the `-dir` option produces the appropriate output if RDF directors exist on such arrays. The `symstat` command produces the appropriate output when issued against a VMAX 10K array with RDF specific options including the `-hop2` option. These options include `-type RDF`, `-RepType rdfa`, `-RepType rdf`, and `-type REQUESTS -RA`.

SRDF/A session statistics by RA group

Statistics for SRDF/A sessions can be monitored by RA group number. To return statistics for all RA group numbers, specify `all`.

```
symstat -sid 179 -i 60 -c 2 -type RDF -rdfg 11

RDF Group Level I/O Statistics:

GRP       IO/sec        MB/sec    % Hits   IO Service Time (usec)    Q
13:33:31 READ   WRITE   READ  WRITE     RD    Min        Max    Avg  Len
13:33:31  11      100    200  12345   34500    30        23       0   10

13:34:31  11      110    220  15000   45000    40        30       0   15
```

SRDF device-level statistics

SRDF statistics can be monitored at the device level. Device-level statistics can be obtained by specifying a device name (`-dev` *`SymDevname`*), logical device name (`-ld` *`LDevname`*), or device group name (`-g` *`DgName`*).

Note: A device group name (`-g` *`DgName`*) must be specified when a logical device name (`-ld` *`LDevname`*) is used.

For example, to return SRDF device-level statistics for device 1085, enter:

```
symstat -i 5 -sid 345 -dev 1085 -type RDF
RDF Device Level I/O Statistics:

         Sym     RDF        IO/sec        MB/sec    %Hit  IO Service Time(usec)
08:38:20 Dev   GRP MODE   READ  WRITE   READ  WRITE   RD    Min       Max   Avg

08:41:46 01085 162   R2      0      0    0.0    0.0    0   1015      1015  1015
```

SRDF statistics by director number

SRDF statistics can be monitored at the SRDF director level. Director-level statistics can be obtained by specifying a specific director number or specifying `all`.

For example, to return SRDF director level statistics, enter:

```
symstat -sid 179 -i 60 -c 2 -type RDF -dir 1d

RDF Device Level I/O Statistics:

         DIR       IO/sec        MB/sec    % Hits   IO Service Time (usec)    Q
13:37:31        READ   WRITE   READ  WRITE     RD    Min        Max    Avg  Len
14:08:34  32     100     200  12345  34500     30     30          0     10
14:08:34  40     120     220  24567  55432     25     40          0     12

14:09:34  32     110     210  15000  45000     40     50          0     15
14:09:34  40     130     230  26265  61250     35     55          0     20
```

SRDF link-level statistics (Enginuity 5876 or lower)

SRDF director-level statistics can be further enhanced to return throughput information of the SRDF links. To return link-level details, specify the `-rdflink` option as shown in the following example.

```
symstat -sid 179 -i 60 -c 2 -type RDF -dir 2c -rdflink

RDF Link Level I/O Statistics:

          LINK      IO/sec       MB/sec     % Hits   IO-Service-Time(usec)    Echo
14:18:27          READ   WRITE   READ  WRITE     RD    Min    Max    Avg       Delay
14:18:29    0      100    200   12000  21000     35     30      0     50
14:18:29    1      110    210   13000  22000     40     40      0     60
14:18:29    2      120    220   14000  23000     45     50      0     70
14:18:29    3      130    230   15000  24000     50     60      0     80

14:19:29    0      110    210   23000  31000     45     35      0     60
14:19:29    1      120    220   24000  32000     50     45      0     70
14:19:29    2      130    230   25000  33000     55     55      0     80
14:19:29    3      140    240   26000  34000     60     65      0     90
```

Note: The `-rdflink` option is not supported on arrays running HYPERMAX OS 5977 or higher.

CHAPTER 16
Change Tracker (Enginuity 5876 or lower)

This chapter introduces the Change Tracker SYMCLI (`symchg`) command and explains how to monitor array data objects for change within your storage environment.

- Overview 502
- Monitoring data objects for changes 504
- Generating reports of changes 509
- Using Change Tracker for SRDF capacity planning 513

Overview

IMPORTANT

Change Tracker is not supported on arrays running HYPERMAX OS 5977 or higher.

The EMC Solutions Enabler Change Tracker software measures changes to data on an array volume or a group of volumes. The Change Tracker command (`symchg`) is used on FBA devices to mark selected areas of disk storage so that the logical volume occupying those areas can be monitored for change.

Change Tracker uses DeltaMark bitmap technology (formerly called the Symmetrix Differential Data Facility) to identify logical blocks that were changed on a FBA device. Before change tracking can begin, a Change Tracker (DeltaMark) session must be created using the `symchg create` command. The `symchg mark` command is then used to perform a timestamp and mark the selected area of disk storage occupied by a data object using a DeltaMark bitmap.

This allows a marked area to be examined at a later time to view changes to the stored object. The tracked changes can also be saved to a log file for later analysis. Change Tracker data is often used to analyze and design TimeFinder and SRDF configurations.

The array stores information about track changes internally, and makes this information available to applications through a DeltaMark bitmap vector or table. Each DeltaMark bitmap contains a bit for every track on the monitored volume. When a write to the volume alters a track of information, the bitmap is updated to reflect that the track has been changed. (Subsequent changes to that same track have no effect on the bitmap.) Figure 14 illustrates a DeltaMark bitmap table.

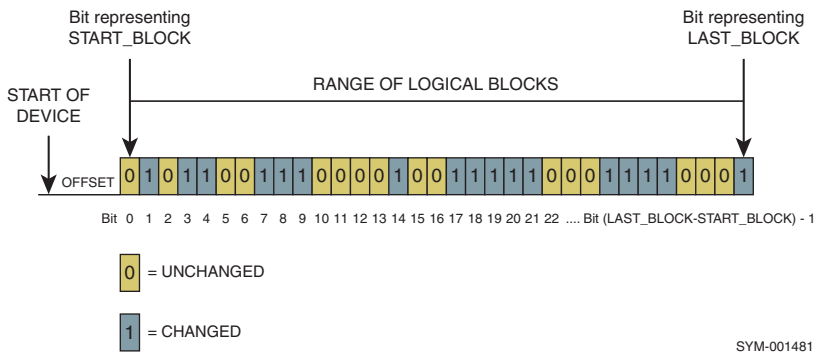


Figure 14 A DeltaMark bitmap table

Change Tracker sessions are allowed on count key devices (CKD) starting with Enginuity 5773. You can create, mark, view, list, and delete a Change Tracker session on CKDs.

The `symchg` command provides full support to non-geometry limited encapsulated devices or external provisioned devices with Enginuity 5876.

Practical uses

Using TimeFinder and SRDF software to mirror data from a source array to a remotely connected array located across the hall or across the globe creates a global data storage solution. That is, production data can be available at a remote site for data mining,

e-business content delivery, application development, testing, backups, and numerous other uses. Moreover, SRDF configurations provide a recovery solution for component or site failures between remotely mirrored devices.

With the wide acceptance of SRDF for remote storage and as a disaster tolerance and disaster restart tool, assessing incremental changes to data over discrete periods of time is a necessary part of planning for data transfer and storage at remote sites. Change Tracker assists in this task by measuring volume changes, providing information so that informed decisions can be made about the communication line bandwidths and links required for such configurations.

Change Tracker command summary

The Change Tracker command (`symchg`) allows you to monitor selected disk storage objects using DeltaMark bitmap technology to track changes and generate reports for later analysis. Change Tracker can monitor individual devices, standard or BCV devices in a device group or composite group, devices in a storage group, or devices listed in a device file.

Note: Change Tracker control operations using the `symchg` command support thin devices (TDEVs) on arrays running Enginuity 5874.210.168 through 5876.

Table 31 summarizes the SYMCLI `symchg` commands you can use to create a Change Tracker session, mark areas to be monitored, remove or delete a session, and generate a report.

Table 31 Change Tracker command summary (page 1 of 2)

Command	Argument	Displays
`symchg`	`create`	Creates a Change Tracker session for the disk device containing the disk storage object to be monitored.By including the `-type` option in the command, you can create a write (default), read, or read/write Change Tracker session. Only one session can exist on a device at a time. For example, you cannot create a read session on a device that already has a write session on it.
	`delete`	Unmarks the disk storage object and deletes the Change Tracker session.
	`list`	Lists all devices marked as Change Tracker objects, including object name and type, session type, and timestamp of marking. The`-type` option lists write, read, or read/write sessions.
	`mark`	Marks disk storage objects as unchanged so that new changes can be tracked. Requires that a Change Tracker session first be created.

Table 31 Change Tracker command summary (page 2 of 2)

Command	Argument	Displays
symchg	remove	Unmarks the disk storage object while retaining the Change Tracker session.
	report	Generates a report on the amount of change for the specified object in a given log file. Reports can be generated for device groups, composite groups, and device files.
	view	Displays the amount and rate of change for marked disk storage objects or shows if a Change Tracker session exists for a specified disk storage object.

For information about performing Change Tracker control operations, refer to "Monitoring data objects for changes" on page 504. For details about the `symchg` command syntax and other SYMCLI commands, refer to the *EMC Solutions Enabler SYMCLI Command Reference Guide*.

Using -force with delete and remove

The `symchg remove` does not remove devices from the Change Tracker database if the object type does not match the object type in the command. Additionally, if the `symchg remove` command fails half way through its device list, you must issue a remove on each device that was not removed. In this case, you can use the `-force` option to force the removal of all the devices from the database.

You can also use the `-force` option with `symchg delete` to force the deletion of the Change Tracker session from an array and the deletion of the devices from the Change Tracker database.

Monitoring data objects for changes

To mark areas of array disk storage for change monitoring, you first create a Change Tracker session for the objects that you want to monitor. Creating a Change Tracker session creates DeltaMark bitmaps for the specified objects. Once a session has been established, the objects can be marked at any particular point in time, which resets the bitmaps for those objects (that is, marks all tracks as unchanged). Marking the objects is the starting point from which changes will be measured.

Monitoring the objects is accomplished by viewing the objects in a session. Viewing objects processes the bitmaps that correspond to the objects and displays the amount and rate of change. Viewing can be performed at discrete intervals of time and for a particular number of executions, and the bitmaps can also be reset after each sample. The display may be saved to a user-defined log file, where it is stored in a comma-separated format, enabling the data to be imported into a spreadsheet for analysis. A summary report can be produced from the results in a log file, and the log file can be used to perform capacity planning by specifying particular variables and calculating other parameters.

The `symchg` command operates on disk storage objects by specifying an array device name, a logical device name, a device group, a composite group, or a device file. Individual objects may be removed from a session without terminating that session (so other objects in the same session continue to be monitored), and a session can be deleted, terminating the monitoring of all objects included in that session.

Note: SYMCLI `symchg` only monitors FBA devices. You can monitor CKD devices using the mainframe change tracker utility.

Change Tracker does not work correctly if the configuration of a metadevice changes. Once a Change Tracker session is created, the following is blocked through `symconfigure`:

- Non-meta device becomes a metadevice.
- Changing of metamembers.
- Changing a metadevice from a concatenated meta to a striped meta.
- Meta device dissolves and is no longer a metadevice.

Change Tracker does not allow you to create a session on a thin device that is not bound. Therefore, once a device has a Change Tracker session defined on it, `symconfigure` does not allow you to unbind that device.

Creating a Change Tracker session

Before marking objects, you must create a Change Tracker session for the device(s) containing the objects by using the `symchg create` command.

Use the following syntax to create a Change Tracker session:

```
symchg [-sid SymmID] [-v]
   create dev SymDevName [-type read | write | rw]

symchg -g DgName | -cg CgName [-bcv] [-v]
   create [ld LdevName] [-type read | write | rw]

symchg -sg SgName -sid SymmID [-v]
   create [-type read | write | rw]
symchg -file DevFile -sid SymmID [-v]
   create [-type read | write | rw]
```

Setting the session type

The `symchg create` operation has a `-type` option to indicate the type of session to create for the devices in a Change Tracker object. This feature is available with Enginuity 5875 and 5876.

The sessions are:

- Write (W): determines how much data was written to a device. Many write sessions can be opened on a device. This is the default.
- Read (R): identifies a protection bitmap of the tracks read by a host. Only one read session can be opened on a device.
- Read/Write (RW): can perform both a read and a write on a device.

For example, to create a Change Tracker session of type write for a single device (17C) on array ID 55, enter:

```
symchg create dev 17c -sid 55
```

Since the write session type is the default, you do not have to specify a type.

You can mark and examine more than one object per session and you can specify a device name (*SymDevName*) or a logical device (*LdevName*) name within a device group or composite group for the session. You can specify the standard or BCV device (-bcv) of a device group or composite group.

Creating a device text file

You can create a Change Tracker session for devices listed in a device text file. The device file syntax contains a single column of devices specified by the device name (*SymDevName*). Lines in the device file that begin with a pound symbol (#) will be ignored. The device filename (-file *FileName*) will be inserted into the command line for control operations.

A sample of a device text file listing devices is shown in the following example.

```
# dev_file_1
# Symmetrix Device List for Change Tracker session
#
00A
00B
00C
057
058
# End
```

Marking a data object

After creating a session, you can mark the device storage objects for change monitoring using the symchg mark command. This allows objects to be marked from a particular point in time, with all tracks being marked as unchanged. Once marked, the objects can be monitored for changes. To do this, use the following command syntax:

```
symchg [-sid SymmID] [-v]
   mark   dev SymDevName

symchg -g DgName | -cg CgName [-bcv] [-v]
   mark   [ld LdevName]

symchg -sg SgName -sid SymmID [-v]
   mark

symchg -file DevFile -sid SymmID [-v]
   mark
```

Note: You must use the same convention for the mark command as was used with the symchg create command when the session was created. If the session was created using a device group, composite group, storage group, or device file, the devices must be managed using the same method.

When you mark an object, an entry is added or updated for the specified object in the symchg database. The mark command causes the DeltaMark bitmaps to be reset (no changes) for the devices of the specified object.

You can specify a device name (00A) or a logical device name (DEV001) for the object to be marked. You can also mark the BCV devices (`-bcv` option) of a device group or composite group.

For example, to mark device (`17C`) on array ID 000125600055, enter:

```
symchg mark dev 17c -sid 55
```

Listing marked objects

You can use the `symchg list` command to list disk storage objects that were marked and exist in the Change Tracker database.

To list all logical disk storage objects marked for change monitoring, enter:

```
symchg list
```

To list all disk devices marked for a Change Tracker session of a read/write type, enter:

```
symchg list -session -type rw
```

To list all disk devices marked for a Change Tracker session of type read and visible to the host, enter:

```
symchg list -session dev
```

Removing marked objects

You can unmark a disk storage object from the `symchg` database without terminating the Change Tracker session by using the `symchg remove` command. To do this, use the following command syntax:

```
symchg [-sid SymmID] [-v]
   remove dev SymDevName [-force]

symchg -g DgName | -cg CgName [-bcv] [-v]
   remove [ld LdevName] [-force]

symchg -sg SgName -sid SymmID [-v]
   remove [-force]

symchg -file DevFile -sid SymmID [-v]
   remove [-force]
```

Note: You must use the same convention for the `remove` command as was used with the `symchg create` command when the session was created. If the session was created using a device group, composite group, storage group, or device file, the devices must be managed using the same method.

The remove command causes the DeltaMark bitmaps to stop monitoring changes for the devices of the specified object, while maintaining the session.

You can specify a device name (such as, 00A) or a logical device name (such as, DEV001) for the object to be unmarked. You can also unmark the BCV devices (`-bcv` option) of a device group or composite group.

For example, to unmark device (`17C`) on array ID 000125600055, enter:

```
symchg remove dev 17c -sid 55
```

To unmark BCV devices in the device group `ProdDB`, enter:

```
symchg -g ProdDB remove -bcv
```

To unmark devices listed in the device text file `dev_file_1`, enter:

```
symchg -file dev_file_1 remove
```

Deleting marked objects

You can unmark a disk storage object from the `symchg` database and delete (terminate) the Change Tracker session by using the `symchg delete` command. To do this, use the following command syntax:

```
symchg [-sid SymmID] [-v]
   delete dev SymDevName [-force]

symchg -g DgName | -cg CgName [-bcv] [-v]
   delete [ld LdevName] [-force]

symchg -sg SgName -sid SymmID [-v]
   delete [-force]

symchg -file DevFile -sid SymmID [-v]
   delete [-force]
```

Note: You must use the same convention for the `delete` command as was used with the `symchg create` command when the session was created. If the session was created using a device group, composite group, storage group, or device file, the devices must be managed using the same method.

The delete command causes the DeltaMark bitmaps to stop monitoring changes for the devices of the specified object and deletes the session.

You can specify a device name (such as, 00A) or a logical device name (such as, DEV001) for the object to be unmarked. You can also unmark the BCV devices (`-bcv` option) of a device group or composite group.

For example, to unmark device (`17C`) on array ID 000125600055 and delete the session, enter:

```
symchg delete dev 17c -sid 55
```

To unmark BCV devices in the device group `ProdDB`, enter:

```
symchg -g ProdDB delete -bcv
```

To unmark devices listed in the device text file `dev_file_1`, enter:

```
symchg -file dev_file_1 delete
```

Viewing marked objects

You can process the DeltaMark bitmap corresponding to the disk storage object to display the amount and the rate of change by using the `symchg view` command. For example, to display the amount and the rate of change for device (`17C`) on array ID 000125600055, enter:

```
symchg view dev 17c -sid 55
```

Note: You can verify the existence of a Change Tracker session for a specified object by including the `-session` option.

To display the amount and the rate of change for BCV devices in the device group `ProdDB`, enter:

```
symchg -g ProdDB view -bcv
```

Change information for marked objects in a device group, composite group, storage group, or listed in a device file can be saved to a separate log file. The data written in log file is based on 32k track size; optionally, you can choose to return the output in the native track size by specifying the `-native` option instead. For example, to save change information to file `/tmp/logfile` for marked storage objects in device group `ProdDB`, enter:

```
symchg -g ProdDB view -reset -log /tmp/logfile -c 10
```

The `-reset` option resets the DeltaMark bitmaps for each reporting interval and requires using the count (`-c`) or interval (`-i`) options. The reset after each measurement of changed tracks is called the DELTA collection method.

The SUM collection method calculates the cumulative number of changed tracks from the starting point of when the object was first marked. To collect changed tracks for both the SUM and DELTA collection methods at the same time, you can use the `-multi` option.

For example, to save change information to file `/tmp/logfile` for marked storage objects in device group `ProdDB` using both the SUM and DELTA collection methods, enter.

```
symchg -g ProdDB view -multi -log /tmp/logfile -c 10
```

Note: At each specified interval, the log file will capture both the cumulative number of changed tracks and the number of changed tracks per interval.

Generating reports of changes

Change Tracker allows you to generate two kinds of reports—SUM and DELTA. Reports can be generated for devices in a device group, composite group, or devices listed in a device file.

SUM reports

SUM reports track changes from the point-in-time the DeltaMark bitmap was marked at the beginning of change measurement. Subsequent writes to the same track are not monitored, because the bitmap is not reset after every measurement interval. For instance if the initial *mark* action occurred at 6 A.M. and change data was collected each hour for 12 hours, it would be possible to measure the tracks changed between 6 and 7 A.M., 6 and 8 A.M., 6 and 9 A.M. and so on. It would not be possible to measure the changes between 7 and 8, or 8 and 9 using the SUM report. The SUM reports lend themselves to

refresh interval estimations for SRDF/AR installations. Understanding the change measurement at various intervals enables you to determine the amount of SRDF bandwidth required for different SRDF/AR refresh intervals. Refer to the log file entry below.

```
04/10/2006 06:00:11 , 000184600166 , DG , devgrp , 109 , /dev/rdsk/c3t13d2s2 , 0 , 131070 , SUM
04/10/2006 06:00:11 , 000184600166 , DG , devgrp , 10A , /dev/rdsk/c3t13d3s2 , 0 , 131070 , SUM
04/10/2006 06:00:11 , 000184600166 , DG , devgrp , 10B , /dev/rdsk/c3t13d4s2 , 0 , 131070 , SUM
04/10/2006 06:05:45 , 000184600166 , DG , devgrp , 108 , /dev/rdsk/c3t13d1s2 , 5 , 131070 , SUM
04/10/2006 06:05:45 , 000184600166 , DG , devgrp , 109 , /dev/rdsk/c3t13d2s2 , 0 , 131070 , SUM
04/10/2006 06:05:45 , 000184600166 , DG , devgrp , 10A , /dev/rdsk/c3t13d3s2 , 0 , 131070 , SUM
04/10/2006 06:05:45 , 000184600166 , DG , devgrp , 10B , /dev/rdsk/c3t13d4s2 , 0 , 131070 , SUM
04/10/2006 06:10:38 , 000184600166 , DG , devgrp , 108 , /dev/rdsk/c3t13d1s2 , 9 , 131070 , SUM
04/10/2006 06:10:38 , 000184600166 , DG , devgrp , 109 , /dev/rdsk/c3t13d2s2 , 0 , 131070 , SUM
04/10/2006 06:10:38 , 000184600166 , DG , devgrp , 10A , /dev/rdsk/c3t13d3s2 , 0 , 131070 , SUM
04/10/2006 06:10:38 , 000184600166 , DG , devgrp , 10B , /dev/rdsk/c3t13d4s2 , 0 , 131070 , SUM
     A                 B        C     D     E            F           G    H       I
```

The fields in the log file entry are:

A - Date and time that the sample was collected.

B - Array ID.

C - Type of entity (device group).

D - Name of the device group.

E - Device name.

F - Physical device name. If the device has not been host visible, this field would say `Not Visible`.

G - Number of tracks changed since the group was marked.

H - Total number of tracks on device.

I - SUM indicates that the number in column G represents the cumulative tracks changed since the group was marked.

DELTA reports

DELTA reports track changes that have occurred since the last measurement interval. The DeltaMark bitmap is reset after each measurement of changed tracks. Thus, an hourly report from 6 A.M. to 6 P.M. using the DELTA method would identify how many tracks changed during any given one-hour period. This is useful for determining peak change rates for a given interval of data collection.

To assess the bandwidth requirements for an SRDF/AR implementation, you should first collect a SUM report to determine how often you would want to refresh the remote site. Once the determination of the refresh interval has been finalized (T hours), generate a DELTA report using T hours as the collection interval. This will result in identifying the peak number of changed tracks during any T-hour interval. Refer to the log file entry below for an example of a DELTA report.

```
04/10/2006 06:00:11 , 000184600166 , DG , devgrp , 108 , /dev/rdsk/c3t13d1s2 , 16 , 131070 , DELTA
04/10/2006 06:00:11 , 000184600166 , DG , devgrp , 109 , /dev/rdsk/c3t13d1s2 , 8 ,  131070 , DELTA
04/10/2006 06:00:11 , 000184600166 , DG , devgrp , 10A , /dev/rdsk/c3t13d1s2 , 803 ,131070   DELTA
04/10/2006 06:00:11 , 000184600166 , DG , devgrp , 10B , /dev/rdsk/c3t13d1s2 , 549 ,131070 , DELTA
04/10/2006 06:05:39 , 000184600166 , DG , devgrp , 108 , /dev/rdsk/c3t13d1s2 , 24 , 131070 , DELTA
04/10/2006 06:05:39 , 000184600166 , DG , devgrp , 109 , /dev/rdsk/c3t13d1s2 , 48 , 131070 , DELTA
04/10/2006 06:05:39 , 000184600166 , DG , devgrp , 10A , /dev/rdsk/c3t13d1s2 , 1085 131070 , DELTA
04/10/2006 06:05:39 , 000184600166 , DG , devgrp , 10B , /dev/rdsk/c3t13d1s2 , 809 ,131070 , DELTA
04/10/2006 06:10:21 , 000184600166 , DG , devgrp , 108 , /dev/rdsk/c3t13d1s2 , 25,  131070 , DELTA
04/10/2006 06:10:21 , 000184600166 , DG , devgrp , 109 , /dev/rdsk/c3t13d1s2 , 13,  131070 , DELTA
04/10/2006 06:10:21 , 000184600166 , DG , devgrp , 10A , /dev/rdsk/c3t13d1s2 , 561 ,131070 , DELTA
04/10/2006 06:10:21 , 000184600166 , DG , devgrp , 10B , /dev/rdsk/c3t13d1s2 , 541, 131070 , DELTA

          A              B       C     D     E          F            G    H       I
```

See key above for letter designations.

Viewing reports

When viewing the changes for an object of a device group, the output can either be sent to `stdout` or saved to a user-defined log file. The information saved to the log file is stored in a comma-separated format. This allows the data to be imported into a spreadsheet for analysis. The results in the log file can be examined with the following `symchg report` command:

```
symchg [-ra NumRAs] [-rate KB/s]
             [-resync [mmmm|hh:mm]]
         [-start mmddyyyy hh:mm] [-stop mmddyyyy hh:mm]
         [-backend] [-native] [-v]
         [-devs <[SymDevStart:SymDevEnd | SymDevName]>
                <[,SymDevStart:SymDevEnd | SymDevName...]>]

         report -log LogFile -g <DgName | -cg CgName >
            [-bcv]

         report -file DevFile -log LogFile [-sid SymmID]
```

Each report in a log file shows the amount of change for a specified object of a device group. For more information about the syntax of `symchg`, refer to the *EMC Solutions Enabler SYMCLI Command Reference Guide*.

Report options

Table 32 lists the command line options available for generating a report using the `symchg report` command.

Table 32 SYMCLI report options (page 1 of 2)

Option	Description
`-backend`	Displays information about the DA and back-end disk address of marked devices.
`-bcv`	Generates a report for the BCV devices in the device group or composite group.
`-cg`	Specifies a composite group on which to report.
`-devs`	Specifies a set of array device ranges and/or individual devices for the report action.
`-file`	Specifies a file containing a device list with one *SymDevName* per line.

Table 32 SYMCLI report options (page 2 of 2)

Option	Description
`-g`	Specifies a device group on which to report.
`-log`	Specifies the log file (`-log LogFile`) to read from.
`-native`	Reports data based on native track size.
`-sid`	Specifies the 12-digit ID of an array.
`-sg`	Specifies a storage group on which to report.
`-start`	Defines the start point in the log file to begin processing data. Format is *mmddyyyy hh:mm*. Default is the beginning of the log file.
`-stop`	Defines the stop point in the log file to stop processing data. Format is *mmddyyyy hh:mm*. Default is the end of the log file.
`-ra`	Defines the number of RA directors to use when resynchronizing device pairs in the group.
`-rate`	Specifies the transfer rate to use when resynchronizing device pairs in the group. Format is in *Kb/second*.
`-resync`	Specifies the required time frame in minutes, or hours and minutes, for resynchronizing SRDF device pairs in the group. Format is *mmmm* or *hh:mm*.

Example procedure

The following is an example of typical steps to follow when you are saving change information to a user-defined log file and you are generating a report:

1. Create Change Tracker sessions for each standard device in a device group (such as DATABASE_DG), enter:

   `symchg create -g DATABASE_DG`

2. Mark the devices within the device group (DATABASE_DG).

   `symchg mark -g DATABASE_DG`

   This resets the DeltaMark change information to zero for each device in the device group.

3. Save the change information for the devices of the group (DATABASE_DG) to a log file (`/tmp/change_data.log`) by entering:

   For SUM Reports:

   `symchg view -g DATABASE_DG -log /tmp/change_data.log \ -i 60 -c 100`

   For DELTA Reports:

`symchg view -g DATABASE_DG -log /tmp/change_data.log \ -reset -i 60 -c 100`

Note: In this example, there were 100 samples made at an interval of 60 seconds and the DeltaMark change information is reset after each sample.

For combined SUM and DELTA Reports:

`symchg view -g DATABASE_DG -log /tmp/change_data.log -multi -i 60 -c 100`

4. Produce a summary report of the changed track information for each device in the group (DATABASE_DG) by entering:

   symchg report -g DATABASE_DG -log /tmp/change_data.log -v

   The data in the log file is in a comma-separated format, so you could also import it into a spreadsheet to view the data.

5. If you need to provide SRDF capacity planning that determines how long it would take to resynchronize a device group (DATABASE_DG) with RA directors at a transfer rate of 1000 KB/second, enter:

symchg report -g DATABASE_DG -log /tmp/change_data.log -rate 1000 -ra 2

6. If you need to view information for a device group (DATABASE_DG) on your screen showing the log entries, enter:

   symchg view -g DATABASE_DG

7. To list the objects that are marked and being monitored for change, enter:

   symchg list

8. To verify a Change Tracker session has been created for a device group (DATABASE_DG) and that it is marked, enter:

   symchg view -g DATABASE_DG -session

9. To remove the marked object (DATABASE_DG), enter:

   symchg remove -g DATABASE_DG

   This does not delete the sessions for the devices of the group. It simply records the object indicating it is no longer being monitored (from a host view point).

Using Change Tracker for SRDF capacity planning

SRDF maintains a mirror image of data at the device level in arrays located in physically separate sites. The mirroring in SRDF can be accomplished in real-time (Synchronous mode) or near real-time (Asynchronous or Adaptive Copy modes). These modes of operation are described fully in the *EMC Solutions Enabler SRDF Family CLI Product Guide*. The Adaptive Copy mode is especially useful when transferring large amounts of data during data center migrations, consolidations, and in data mobility environments.

When performing periodic SRDF transfers using Adaptive Copy mode, it is important to estimate the rate of change of data on the volumes to be mirrored in order to make informed decisions about communication line parameters. Change Tracker can help in this process by calculating certain SRDF capacity values (the resynchronization time, the number of remote adapters (RAs), and the RA transfer rates) based on the amount of change to monitored objects.

In the SRDF configuration shown in Figure 15, the R1-BCV is used to periodically transmit changed tracks from the primary site to the remote site. When the BCV is established with the standard as a BCV pair, the SRDF link to its R2 target is suspended. When you split the local BCV pair, you can resume the SRDF link, sending the changes to the remote site.

Using Change Tracker to measure the extent of changes to the local BCV pair while the R1-BCV is suspended from its R2 target allows you to make informed choices about link bandwidth and frequency of transmission for the SRDF transfer.

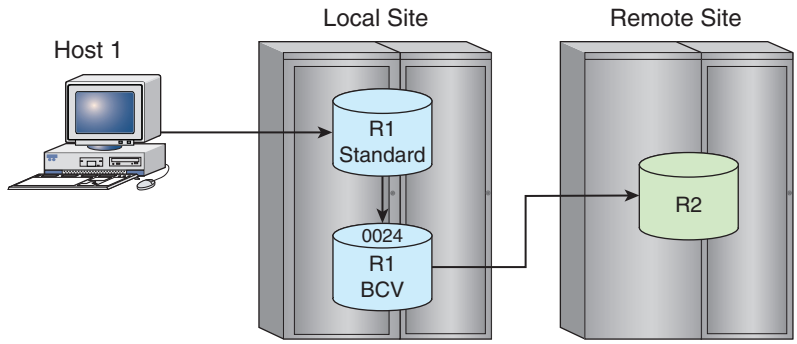


Figure 15 SRDF configuration

The following steps outline an example of setting up devices and using Change Tracker to perform SRDF capacity planning:

1. Identify the device group that you want to monitor. If the group does not exist, you can create it using the `symdg` command. For example, to create a device group named deltadg, enter:

   ```
   symdg create deltadg
   ```

2. Add a device (device name 085 on array number 000000003264) to the device group:

   ```
   symdg -g deltadg -sid 3264 add dev 085
   ```

   By default, SYMCLI assigns a logical device name (of the form DEVnnn) to this device.

3. Create a Change Tracker session for the device group:

   ```
   symchg -g deltadg create
   ```

4. List the sessions that have just been created for the device group:

   ```
   symchg -session list
   ```

5. Mark the device group (this is the starting point from which changes will be measured):

   ```
   symchg -g deltadg mark
   ```

6. In a log file, record cumulative track changes that occur on each device in the device group. This example collects 60 samples at two-minute (120 seconds) intervals in a log file named deltalog:

   ```
   symchg -g deltadg -log deltalog -i 120 -c 60 view
   ```

7. Use the collected data from the log file for SRDF capacity planning by providing two of the following three variables to a `symchg report` command:

   - The number of links
   - The link bandwidth
   - The amount of time available to synchronize the source (R1) and target (R2) devices

   The `symchg report` command calculates the third variable.

CHAPTER 17
Optimizing Array Performance (Enginuity 5876 or lower)

This chapter describes Optimizer concepts and how to improve array performance using the SYMCLI `symoptmz` command.

- Overview 516
- How does Optimizer work? 518
- Performing Optimizer operations using symoptmz 523
- Setting control parameters 524
- Setting advanced parameters 525
- Setting time windows 526
- Setting the swap priority for devices 526
- Setting manual swap lists 526
- Reviewing Optimizer plans 527
- Approving a swap list 533
- Performing rollbacks 534
- Reading Optimizer logs 534
- Migrating devices 535

Overview

IMPORTANT

Optimizer is not supported for arrays running HYPERMAX OS 5977 or higher.

Symmetrix Optimizer improves array performance by continuously monitoring the back-end activity and swapping highly active volumes with idle volumes to achieve improve the balance of the workload across the disks. This automated, continuous process is based on user-defined parameters and is completely transparent to end users, hosts, and applications in the environment.

The Optimizer application (client and server) resides on the service processor, as shown in Figure 16.

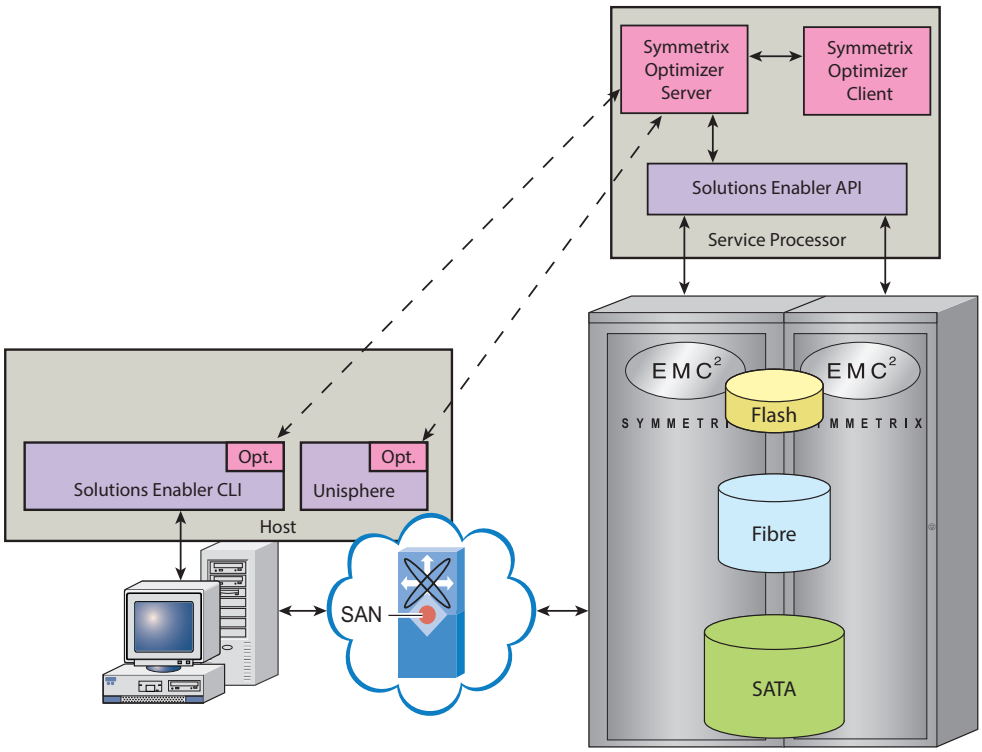


Figure 16 Optimizer application architecture

Optimizer always works within a single array and does not change the protection of a device.

The Optimizer client interface on the service processor is used by EMC Customer Service. From a host system, Solutions Enabler CLI commands and Unisphere dialogs can communicate with the Optimizer server. The SYMCLI `symoptmz` command provides all the necessary commands to control the behavior of Optimizer.

The first part of this chapter provides an overview of the Optimizer functionality, its requirements and restrictions, the different modes of operation, and a description of the process Optimizer uses to make the device swaps.

The second half of the chapter describes the `symoptmz` command, and explains all the options, variables, and parameters that can be used to manage Optimizer on the array.

Data movement

Optimizer provides two types of data movement: full device swap and device move. In a device move operation, a set of source devices will be moved to a list of disks; the array internally determines the device distribution on the disks.

Configuration requirements

This section provides the common requirements, then presents the requirements by Enginuity version.

The following restrictions apply to devices that are considered for a swap:

- Same size and emulation
- Same protection (Enginuity 5874)
- Available DRVs (Dynamic Reallocation Volumes) for both source and target
- Currently attached to legal RAID groups

Optimizer can optimize all open system and mainframe RAID 1 (mirrored), RAID 5, RAID 6, and RDF-protected devices in an array. Optimizer collects statistics for all devices in the array, however, the devices listed below are not supported. This means that statistics are still collected for them, but the devices will not be swapped by Optimizer.

- BCV devices
- CKD meta
- DATA devices
- DRV devices
- Marked by user as "Do not move"
- SFS devices
- Striped CKD devices
- Thin device
- Vault devices
- Virtual / SAVE devices

Enginuity 5874 or 5876

DRV devices

All DRVs are mirrored and only one DRV per swap is required. In addition, the DRV can be equal or larger in size than the logical device that is swapped.

Open mirror position

Arrays running Enginuity 5874 or 5876 do not require the open mirror position.

Open configuration lock requirement

Swaps on arrays running Enginuity 5874 or 5876 do not explicitly prevent other configuration changes from taking place during the swap operation.

Enginuity 5773 or earlier

DRV devices

As a minimum, two DRVs must be configured for each size and emulation of the volume to be swapped by Optimizer. For example, for Optimizer to swap a 4 GB open system volume, then two 4 GB open system DRVs must be configured. If more than one simultaneous swap is desired, then additional DRVs are needed (two for each swap). If there are volumes of mixed size or emulation within the same array, then DRVs are required for each size and type to be swapped.

Open mirror position

Optimizer requires an open mirror position to be able to perform a swap, however, arrays (running Enginuity 5773 or earlier) have a four mirror slot limitation per volume. This limitation may make it difficult in certain configurations for Optimizer to obtain the mirror position it requires to perform a swap. For example, a RAID 1 device already has two BCVs established, and Optimizer wants to swap this device, it cannot get a mirror slot for the DRV. If Optimizer does not have a mirror slot available, then it will drop the swap and rerun the analysis. If Optimizer is in Rollback, Manual, or User-Approved mode, then Optimizer will retry a number of times until an error is encountered. Eventually, Optimizer will stop trying if a mirror slot does not become available.

Configuration lock requirement

For arrays running Enginuity 5773 or earlier, Optimizer holds a Symmetrix External Lock (SEL) to perform a swap or migration. The configuration lock is an exclusive lock on the array for performing configuration changes. This lock prevents multiple applications from changing the array configuration at the same time. If another application holds the configuration lock when Optimizer wants to create a swap, then Optimizer will behave as previously described (no open mirror position). Optimizer will hold the configuration lock for the duration of the swap or migration, or until the operation is canceled, so other applications will not have the ability to perform configuration changes during this time.

Note: Committed swap or migration operations that are scheduled for a later time can be canceled up until the scheduled time.

How does Optimizer work?

Once initialized with the user-defined parameters, Optimizer operates totally autonomously on the array service processor. Optimizer monitors and controls array performance by following these steps:

Step 1: Optimizer builds a database of device activity statistics on the array back-end. This data is saved for a maximum of two weeks.

Step 2: Using the statistical data collected, configuration information, and the user-defined parameters, the Optimizer algorithm identifies busy and idle devices and their locations on the physical drives. The algorithm tries to minimize average disk service time by balancing I/O activity across physical disks. Optimizer determines which disks require balancing by locating busy devices close to each other on the same disk, and/or by locating busy devices on faster areas of the disks. Optimizer takes into account the speed of the disk, the disk geometry, and the actuator speed to determine faster disks.

Step 3: Once a solution for load balancing has been developed, the next phase is to carry out the array device swaps. You can specify whether swaps should occur in a completely automated fashion, or if the device swaps require user approval before the action is taken.

Step 4: Once a swap function completes, Optimizer continues data analysis for the next swap.

Performance metrics

Optimizer looks only at back-end activity; it uses the back-end logical device statistics, and the following metrics:

- DA logical volumes reads
- DA logical volumes writes
- Logical volume prefetch
- DA logical volumes KB read
- DA logical volumes KB written

While modeling the disk service time, different weights are assigned to read, write, and prefetch activity. Optimizer assumes that reads are equally spread among all the device mirrors, and all mirrors perform all writes.

Modeling service time

Accurate seek and latency times are impractical to get because they require a complete trace of I/Os sequence, therefore the Optimizer uses a mathematical model to calculate these metrics. Service time is defined as the sum of seeks, latency and transfer time:

- **Seek time** — Seek time is the time it takes the disk arm to move and position the disk head on the correct track—move from track *X* (serving the previous I/O) to track *Y* (serving the next I/O). Optimizer uses a gig-to-gig database to model seek time for different addresses and different disk drives.
- **Latency time** — Latency time is the delay for disk rotation. The latency time is a function of the disk rotational speed. Optimizer assumes one-third of a spin for each I/O, assuming the internal disk optimization is on.
- **Transfer time** — Transfer time is the time that it takes the disk to transfer the data from/to the disk. Transfer time is a function of the data transfer rate, disk bandwidth, and data layout. Optimizer uses the Zone Bit Recording (ZBR) database to model the transfer time of data. The ZBR database includes information about the bandwidth of each zone of the disk.

Determining device swaps

When Optimizer evaluates the performance statistics, it determines potential device swaps based on how well they would improve overall performance. This analysis is based on minimizing disk service time (rotational latency plus seek time plus transfer time). Optimizer uses three strategies when determining which swaps to make.

1. **Load balancing**—Swaps highly active devices on disks with lower activity to even out the load across the physical drives in the system. This decreases the contention on the individual physical disks, improving the performance of both highly active devices and low-activity devices.
2. **Minimizing seek**—Performs swaps that relocate highly active devices so that they are closer together on the physical disk and will decrease the seek distance for I/Os to these devices. When Optimizer swaps devices, it will try to achieve this type of configuration since it decreases the overall I/O service time.
3. **Using faster media**—Optimizer tries to swap highly active devices on the outer zones of the disk. This is because devices located on the outer zones of the disk have faster transfer speeds.

Optimizer's algorithm uses detailed disk performance information that takes into account several drive characteristics, such as those gig-to-gig seek times, zone-bit recording, and bandwidth data.

Finding the best swap

The Optimizer analysis consists of three high-level phases:

- **Calculate service time** — Model and sum the total service time of each disk and for every time stamp that was marked to be included by the analysis.
- **Sort disks by activity** — Sort all disks by their modeled total service time.
- **Find best swap** — Starting at the busiest disk, check all potential swaps. The analysis process models *what-if* scenarios using *virtual swaps* to estimate the impact of a swap on the service time of the affected disks. Optimizer checks as many swaps as possible in order to guarantee that the best swap is indeed selected.

For Enginuity 5773 and earlier, 1 to 256 hypers are allowed per disk. The maximum number of hypers per physical disk automatically sets to 512 for Enginuity 5874 and 1024 for Enginuity 5876. Finding a swap that improves service time by 10 percent or more on these systems is very rare. Since using a percentage of improvement as swap *goodness* criteria is not applicable anymore, Optimizer had to adapt a different method (*M0*) that deals better with the "many hypers contribute smaller chunks" problem. M0 is defined as "the best you could possibly get" from a swap. Each disk in the system is assigned an M0 designation, which is defined as the minimum service time a disk can get by replacing one of its hypers by a null hyper (a hyper that performs no I/Os to the disk).

When analyzing a swap, Optimizer checks how close the new-modeled service time is to the disk's M0; the closer to M0, the better the swap. Usually, two disks are affected by a single swap; the busier disk's service time is expected to go down, while the other disk's service time is expected to go up. In addition to the M0, the Optimizer also ensures that the new maximum of service times is less than the old one.

Swap procedure for Enginuity 5874 -5876

Arrays with Enginuity 5874 or 5876 streamline the device swap procedure by using only one DRV device during the swap. With these Enginuity levels, only mirrored DRV devices can be created. Refer to Table 11 on page 217 for information about creating mirrored DRV devices.

Note: Refer to "Swap procedure for Enginuity 5773 and earlier" on page 521 if you have an earlier version of Enginuity.

Hypervolumes are swapped using a four-step process. One DRV must be configured per swap. The DRV should be of equal or greater size that the swapped devices. With Optimizer, up to eight simultaneous swaps can happen at one time.

Figure 17 on page 521 illustrates the steps for swapping device A with device B.

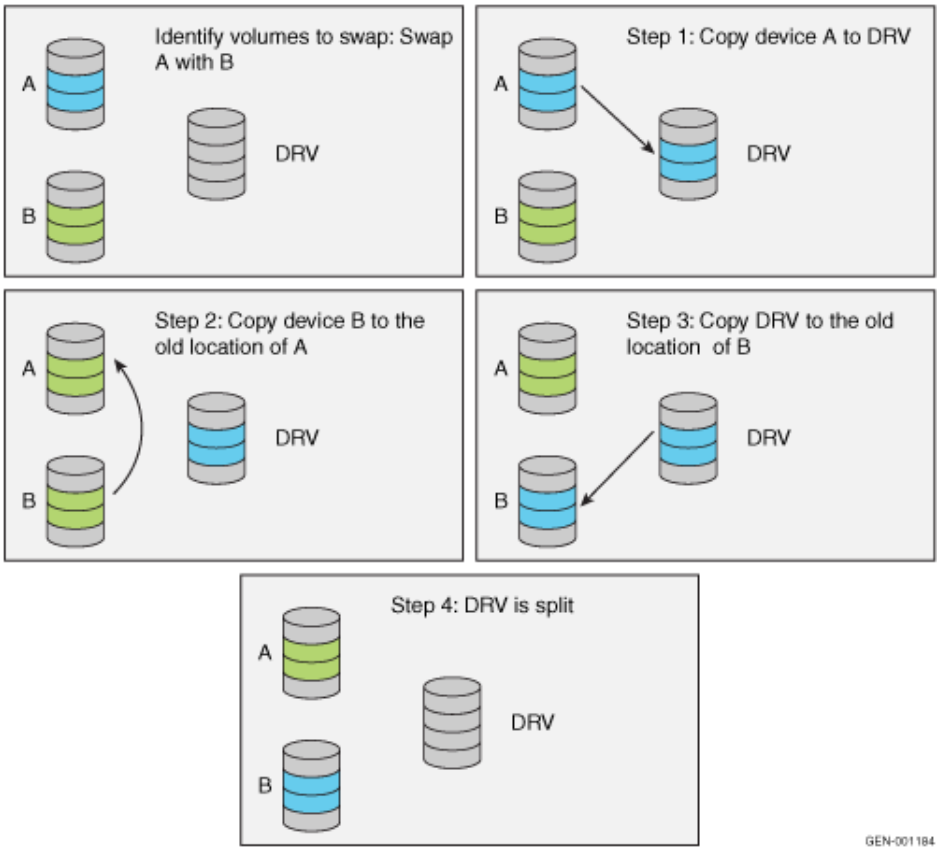


Figure 17 Swap steps using one DRV

Swap procedure for Enginuity 5773 and earlier

The swap procedure relies on TimeFinder technology and uses DRVs as temporary mirrors. The following section describes the four swap steps when dealing with mirrored devices swaps. In the case of three mirror devices, the swap procedure does not require a DRV.

Step 1: Identify volumes to swap

Optimizer identifies a pair of hypervolumes to swap by recognizable patterns of hypervolume activity and criteria. In Figure 18, assume the red volumes (on A) have high activity and the blue volumes (on B) have low activity.

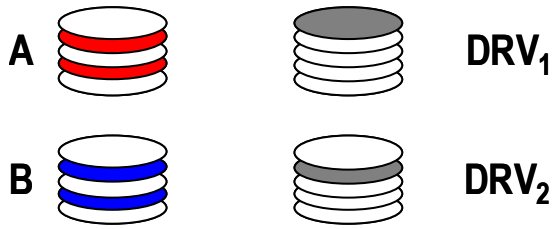


Figure 18 Identify volumes to swap

Step 2: Copy volume to DRV

Optimizer swap commands are passed to SymmWin, which assigns one DRV (DRV1) as a third mirror for hypervolume A. A second DRV (DRV2) will be assigned as a third mirror for hypervolume B. All tracks on the third mirror are marked invalid. Tracks are copied from the valid mirrors to the two DRVs. After the DRVs are synchronized, the two original swap physical mirrors are marked Not Ready (volume A and volume B) and their attributes are swapped (see Figure 19). Both hypervolumes still have two (or more) physical mirrors. Host activity to the hypervolumes is now directed to DRVs and the other mirror.

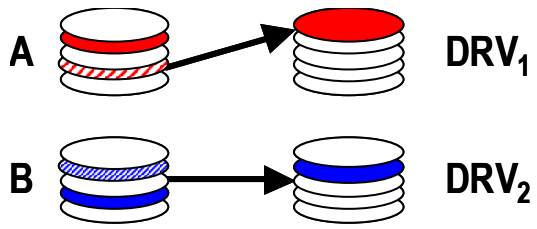


Figure 19 Copy volume to DRV

Step 3: Copy DRV to new location

The data on the DRVs are now copied to the new location. After the attributes of the original hypervolumes are swapped, SymmWin copies the tracks from the valid mirrors to the two new mirrors and then makes the original hypervolumes Ready (see Figure 20). This is similar to a BCV restore.

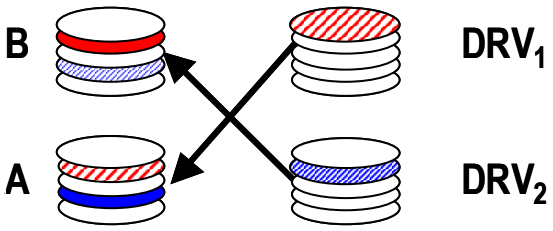


Figure 20 Copy DRVs to new locations

Step 4: DRVs are split

The final step is to split the DRVs from their standard and mirror hypervolumes after synchronization completes (see Figure 21). The drive's balance improves and the DRVs are now available for the next swap.

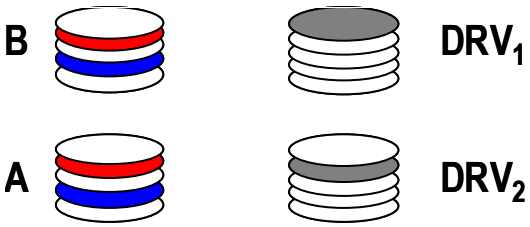


Figure 21 DRVs are split

Note: Swap of three mirror devices does not require a DRV.

Legality of swaps

As a rule, Optimizer will never suggest a swap that conflicts with configuration rules as defined by the Enginuity, SymmWin, or the configuration groups. To minimize the affect of the swap procedure on the overall array performance, the Optimizer will not include the same spindle twice in one swap group. In addition, you can use QoS controls to set a lower BCV priority for the STDs to be swapped, and plan the swap activities to occur on idle business time using the Time Window feature. Chapter 12, "Managing Quality of Service" explains how to use QoS controls.

Performing Optimizer operations using symoptmz

SYMCLI on the host system issues commands to display the values of the parameters that control the behavior of the Optimizer. Some of the `symoptmz` command actions are executed directly on the command line, while others are processed in a command file.

Table 33 describes the `symoptmz` command actions that can be executed on the command line.

Table 33 Optimizer CLI command actions (page 1 of 2)

`symoptmz` command argument	Description
`clear_stats`	Clears disk statistics maintained by Optimizer.
`commit`	Updates Optimizer with the changes defined in the command file.
`disable`	Disables the Optimizer algorithm processing.
`enable`	Enables the Optimizer algorithm processing.
`list`	Displays Optimizer-specific attributes of array devices.
`prepare`	Performs extra range checks on the changes specified in the command file.
`preview`	Verifies the syntax of the changes specified in the command file.
`query`	Queries Optimizer and displays the current state and version information of the Optimizer. If `-v` is specified, additional information about the version and open sessions displays.
`read`	Reads from Optimizer's log file.
`rollback`	Rolls back an array configuration to that of a previous point-in-time.

Table 33 Optimizer CLI command actions (page 2 of 2)

symoptmz command argument	Description
show	Shows information about the current Optimizer data movement/swap plans or configuration parameters.
sync	Acquires version information from Optimizer. This argument is supported only in conjunction with -version.

Enabling and disabling Optimizer

To start and stop the Optimizer process on the array service processor, use the symoptmz enable and symoptmz disable commands. When enabled, Optimizer collects data and performs data movement. When disabled, Optimizer does not perform data movement.

Committing command file changes

Commands for setting or clearing any parameters or time windows may be placed in a command file, which will then be processed by symoptmz. Alternatively, stdin redirection can be used with "here documents" in UNIX shell scripts.

Each command in the file has to be terminated by a semi-colon (;). There is no limit on the number of commands or the type of commands that can be placed in a command file. All the commands in a command file are executed in a single Optimizer session. The commands in the command file are not case sensitive, however, the parameters entered are case sensitive.

To change any Optimizer parameters, use a command file with the commit option.

Example To commit the changes specified in the command file opt_config.txt, issue the command:

```
symoptmz -file opt_config.txt commit
```

The commit option carries out syntax and range checks, and then updates the Optimizer with the modified parameters.

Clearing the Optimizer statistics

To clear the disk statistics maintained by Optimizer, use the following form:

```
symoptmz -sid SymmID clear_stats
```

Setting control parameters

Note: Control parameters can be set for Optimizer or FAST. Both tools use the same command file format.

Optimizer control parameters are set using the following syntax in the command file:

```
set control_parms [start_mode=AUTO | MANUAL>,]
  [swap_mode=AUTO | USER_OK,]
  [min_perf_period=min_perf,]
  [workload_period=workload,]
  [max_simult_swaps=max_simult,]
  [swap_rate=max_swaps];
```

Table 34 describes each parameter.

Table 34 Optimizer control parameters

Parameter	Description
start_mode	Determines whether Optimizer is enabled or disabled whenever Optimizer is launched. For example, after rebooting the service processor.
swap_mode	Controls whether Optimizer should automatically swap devices (`AUTO`) as soon as it finds swaps that would improve performance. If Optimizer is enabled in `AUTO` swap mode, each day it will make up to the number of swaps specified by the `swap_rate` parameter, under control of the swap time window settings. If Optimizer is enabled in `USER_OK` swap mode, it will generate lists of swap suggestions approximately once an hour, and then wait for the user to approve the swap list before proceeding.
min_perf_period	Specifies the amount of samples required initially before a recommendation will be made. You should make sure that the values you specify are long enough (usually a week) for Optimizer to establish a good characterization of your typical workloads. This parameter is expressed in hours. Keep in mind that the Optimizer statistics database holds about 14 days worth of data.
workload_period	Specifies how far back in time Optimizer should consider when the optimization algorithm is run. Be careful not to make this value too large or you may include data that is so old it is no longer representative of your current workloads. This parameter is expressed in hours.
max_simult_swaps	Controls how many swaps Optimizer can perform simultaneously (up to four). The actual value should reflect the number of devices that will be swapped simultaneously. The acceptable values are from two (a single pair swapped) to eight (four pairs swapped) Optimizer swaps.
swap_rate	Sets the maximum number of swaps that Optimizer is allowed to make in a single day. This parameter is only relevant if `swap_mode` is set to `AUTO`. Note that reflects the total number of devices to be swapped in a day. A value of 24 would allow 12 pairs of devices to be swapped in a day.

Example The following command file sets up Optimizer to analyze data from the previous seven days and to start figuring out swap suggestions after three days of collecting data. It also sets Optimizer to User Approved mode and sets the maximum number of simultaneous swaps to eight (four pairs of hypervolumes):

```
set control_parms start_mode=AUTO,
  min_perf_period=72,
  workload_period=168
  swap_mode=USER_OK,
  max_simult_swaps=8,
  swap_rate=50;
```

Setting advanced parameters

Optimizer provides the following advanced parameters that can be set:

```
set advanced_parms [max_rollback = <max_days>,]
                   [hot_spot = <TRUE | FALSE>,];
```

Where:

max_days — Specifies the maximum number of days for rollbacks. For example, if 8 were specified, only swaps that took place within the past eight days would be eligible for rollback.

`hot_spot` — For each time sample, the disk access time is computed from the statistics collected.

Setting time windows

The `symtw` command manages time windows. You can use this command to convert any existing time windows to the new format. The time window format is for FAST, FAST VP, and Optimizer.

Refer to Chapter 18, "Managing Time Windows (Enginuity 5876 or lower)" for information about managing time windows with the symtw command.

IMPORTANT

You should use the `symtw convert` command to convert any existing time windows (previously created with the `symoptmz` command) to the new format.

Setting the swap priority for devices

There may be devices that store critical application data for which you want the highest possible performance, or devices that you never want Optimizer to swap.

Use the following command to set the device priority:

```
set swap_priority <NO_SWAP | NORMAL | HIGH> for
    dev SymDevStart[:SymDevEnd];
```

Where:

NO_SWAP — sets the device to never be swapped.

NORMAL — allows swaps that will improve performance.

HIGH — requires the best performance.

Example To set the swap priority for device `020` to `HIGH`, commit a command file containing the following:

```
set swap_priority HIGH for dev 020;
```

Setting manual swap lists

Enginuity 5874 and 5876

To set a manual swap list, use the following form:

```
set dev_swap SymDevName1 with SymDevName2
  [, SymDevName3 with SymDevName4, ...]
  [begin_at=TimeVal];
```

Example

```
set dev_swap 0030 with 0040;
```

Note: You can set manual swaps between different device protection types.

Enginuity 5773 and earlier

To set a manual swap list, use the following form:

```
set swap_list {Hyper1} with {Hyper2} [, {Hyper3} with {Hyper4},... ]
[begin_at=TimeVal];
```

Where:

Hypern — Is of the form `DDD,I,T,HH`:

DDD — Is the director identifier
I — Is the director interface
T — Is the target ID
HH — Is the hyper number

TimeVal — Is in the form of `MMDDYYYY:HHMMSS`.

Reviewing Optimizer plans

This section provides examples of the Optimizer-generated plans for swaps.

Show parameters

To show Optimizer's planned activities, use the following arguments with the `symoptmz` command:

```
symoptmz show [-v] -swap_list | -activity_list
          [-manual | -generated ]

       show -parms [-dp | -vp] [-offline]
       show -composite [-dp | -vp]
       show [-v]  -swap_list | -rollback_list
```

Where:

`-activity_list` — Displays all the swap lists currently known to Optimizer (interchangeable with `-swap_list`).

`-generated` — Displays Optimizer-generated swap plans (and FAST-generated plans).

`-manual` — Displays any user-defined swap plans.

`-parms` — Displays information about the control parameters of Optimizer.

`-dp` — Filters the display to include only standard devices. (Disk group provisioning)

`-vp` — Filters the display to include only thin devices. (Virtual provisioning)

If neither filter is specified, all data movement time windows display.

`-offline` — Displays the control parameters and time windows from the cached SYMAPI database.

`-composite` — Displays composite time windows. Composite time windows are generated by Optimizer by combining all known user-defined time windows.

`-dp` — Filters the display to include only standard devices.

`-vp` — Filters the display to include only thin devices.

If neither filter is specified, all data movement time windows display.

`-rollback_list` — Displays a list of possible rollback points.

`-swap_list` — Displays all the swap lists and migrations currently known to Optimizer.

Showing control parameters

Example For example, to show the Optimizer parameters set for array 234 for thin devices only, enter:

```
symoptmz -sid 234 show -parms -vp
```

Output similar to the following displays:

```
Optimizer Control Parameters

  . . .

  Number of Swap Time Windows : 1


    Time Window ID            : The Default Time Window
    Type                      : Swap
    Provisioning              : Thin
    Flags                     : Exclusive
    Periodicity               : Once
    Start Date                : None
    Stop  Date                : None

  Performance Time Windows    : 1

    Time Window ID            : The Default Time Window
    Type                      : Performance
    Flags                     : Inclusive
    Periodicity               : Once
    Start Date                : None
    Stop  Date                : None
```

Example To show the Optimizer parameters set for array 234 for all devices, enter:

```
symoptmz -sid 234 show -parms
```

Output similar to the following displays:

```
Optimizer Control Parameters

  . . .

  Number of Swap Time Windows : 2


    Time Window ID            : The Default Time Window
    Type                      : Swap
    Provisioning              : Standard
    Flags                     : Exclusive
    Periodicity               : Once
    Start Date                : None
    Stop  Date                : None

    Time Window ID            : The Default Time Window
    Type                      : Swap
    Provisioning              : Thin
    Flags                     : Exclusive
```

```
    Periodicity                : Once
    Start Date                 : None
    Stop  Date                 : None

  Performance Time Windows     : 1

    Time Window ID             : The Default Time Window
    Type                       : Performance
    Flags                      : Inclusive
    Periodicity                : Once
    Start Date                 : None
    Stop  Date                 : None
```

Example To show all the composite windows for array 234, enter:

```
symoptmz -sid 234 show -composite
```

Output similar to the following displays:

```
Number of Composite Time Windows : 3

  Start Time            Stop Time             Type  Prov.  Flags
  --------------------  --------------------  ----  -----  ------------

  Dec 31 19:00:00 1999  Dec 30 19:00:00 2030  Swap  STD    Exclusive
  Dec 31 19:00:00 1999  Dec 30 19:00:00 2030  Swap  THIN   Exclusive
  Dec 31 19:00:00 1999  Dec 30 19:00:00 2030  Perf  N/A    Inclusive
```

Showing the swap list

The output for the `symoptmz show` command includes the estimated percent of the plan complete and the estimated time to completion.

Example The following command displays the swap list for array 234. The fields in this display are described after the example:

```
symoptmz show -swap_list -sid 234

This operation may take up to a few minutes. Please be patient...

Symmetrix ID                   : 000194900234
Number of Plans                : 2

Plan ID                        : 12222009:130114
Plan Type                      : Manual Defined
Plan State                     : ApprovedWithConfigParameters
Start Time                     : Tue Dec 22 21:30:23 2009
Percent Complete               : 90%
Estimated time to completion   : 00:02:15
Number of Groups               : 1

Group 1:
  {
  Group Attributes             : Optimizer Manual Swap
  Group State                  : InProgress
  Time Started                 : Tue Dec 22 21:30:23 2009
  Time Completed               : N/A

  Percent Complete             : 90%
  Estimated time to completion : 00:02:15

  Swap Pairs (2)
    {
              Source Device                          Target Device
    ------------------------------------  ------------------------------------
```

```
            Dsk                                     Dsk
      Sym   Grp Group Name         Prot       Sym   Grp Group Name         Prot
      ----  --- ----------------   --------   ----  --- ----------------   --------
      0020  004 sata_disks         R5(3+1)    0086  004 sata_disks         R5(3+1)
      0031  004 sata_disks         R1         0055  004 sata_disks         R1
      }
    }

Plan ID                         : 12222009:154359
Plan Type                       : Auto Generated
Plan state                      : InProgress
Start Time                      : Tue Dec 22 20:30:23 2009
Percent Complete                : 5%
Estimated time to completion    : 04:12:30
Number of Groups                : 3

Group 1:
  {
  Group Attributes              : Optimizer Generated
  Group State                   : InProgress
  Percent Complete              : 20%
  Estimated time to completion  : 00:40:12

. . .
```

The terms in the display are explained in Table 35.

Table 35 Fields in Optimizer display (page 1 of 2)

Field	Description
`Plan Type`	Defines where the plan originated. Possible values are: • `Manual Defined` — User defined swap/rollback plan. • `Auto Generated` — Optimizer/FAST auto-generated plan.
`Plan State`	Defines the current state of the plan. The possible values are: • `NotApproved` — The plan is not scheduled. • `ApprovedWithConfigParamaters` — The plan is approved and scheduled according to a set time window. • `ApprovedWithDelay` — The plan is approved and scheduled according to a user-defined start time. • `Validating` — The Optimizer server is validating the manual swap/rollback plan. • `ConfigInProgress` — Data movement is in progress for the plan. • `Aborting` — The plan is in the process of aborting.
`Percent Complete`	Provides an estimate of how much (percent) of the plan has completed.
`Estimated time to completion`	Provides an estimated time to the plan's completion based on the invalid track data and timestamps returned.
`Delayed Start`	Indicates if there is a delay in the plan start time. Possible values are: • `None` — No delay. • `Time` — Displays an additional Start_time field.
`Group Attribute`	Indicates where the plan originated. Possible values are: • Optimizer Manual Swap • Optimizer Manual Rollback • Optimizer Generated • Fast Generated
`Group State`	Displays the state of the group. The possible values are: • Done • InProgress • NotStarted • `Failed` — The execution of the entire group has failed and will be retried.

Table 35 Fields in Optimizer display (page 2 of 2)

Field	Description
`Time Started`	The time the swap started.
`Time Completed`	The time the swap completed.

Example To show the swap list output for array 234 with the verbose (`-v`) option, enter:

```
symoptmz show -swap_list -sid 234 -v

Symmetrix ID                    : 000194900234
Number of Plans                 : 2

Plan ID                         : 12222009:130114
Plan Type                       : Manual Defined
Plan State                      : ApprovedWithConfigParameters
Start Time                      : Tue Dec 22 21:30:23 2009
Percent Complete                : 90%
Estimated time to completion    : 00:02:15
Number of Groups                : 1

Group 1:
  {
  Group Attributes              : Optimizer Manual Swap
  Group State                   : InProgress
  Time Started                  : Tue Dec 22 21:30:23 2009
  Time Completed                : N/A
  Percent Complete              : 90%
  Estimated time to completion  : 00:02:15

    . . .

Plan ID                         : 12222009:154359
Plan Type                       : Auto Generated
Plan State                      : CnfigInProgress
Start Time                      : Tue Dec 22 20:30:23 2009
Percent Complete                : 5%
Estimated time to completion    : 04:12:30
Number of Groups                : 3

  . . .

Group 3:
  {
  Group Attributes              : FAST Generated(Compliance)
  Group State                   : NotStarted
  Time Started                  : N/A
  Time Completed                : N/A
  Percent Complete              : 0%
  Estimated time to completion  : 04:12:30

  Move Devices(6)
    {
    SRC Device(s)               : 0042 0043 0044
    SRC Protection Type         : R6(14+2)
    SRC Storage Group Name      : OraSales
    SRC Tier Name               : N/A
    SRC Disk Group Number       : 3
    SRC Disk Group Name         : fiber_disks
    TGT Protection Type         : R5(7+1)
    TGT Tier Name               : ArchiveDBTier
    TGT Disk Group Number       : 4
    TGT Disk Group Name         : sata_disks
    TGT Disks                   : [16A, D, 1] [01A, C, E] [15C, C, D] [15D, D, A]
```

```
                                  [01C, D, 2] [12A, D, D] [15B, C, E] [07A, C, 2]
                                  [01D, C, 5] [12C, C, 1] [15B, D, 1] [07A, D, 8]
                                  [01F, C, 5] [12H, C, 1] [07H, D, 1] [07F, D, 8]
    Percent Complete            : 0%

    SRC Device(s)               : 0059 005A 005B
    SRC Protection Type         : R5(3+1)
    SRC Storage Group Name      : OraSales
    SRC Tier Name               : PrimeDBTier
    SRC Disk Group Number       : 2
    SRC Disk Group Name         : flash_disks2
    TGT Protection Type         : R5(3+1)
    TGT Tier Name               : WorkDBTier
    TGT Disk Group Number       : 3
    TGT_Disk Group Name         : fiber_disks
    TGT Disks                   : [16C, D, 1] [01B, C, E] [15A, C, D] [15C, D, A]
                                  [01A, D, 2] [12B, D, D] [15C, C, E] [07B, C, 2]
                                  [01D, C, 5] [12C, C, 1] [15B, D, 1] [07A, D, 8]
    Percent Complete            : 0%
    }
  }
```

Example To display a list of Optimizer-generated plans for array `234`, enter:

```
symoptmz show -swap_list -sid 234 -generated
```

Output similar to the following displays:

```
This operation may take up to a few minutes. Please be patient...
Symmetrix ID                    : 000194900234

Plan ID                         : 12222009:154359
Plan Type                       : Auto Generated
Plan State                      : InProgress
Start Time                      : Tue Dec 22 20:30:23 2009
Percent Complete                : 5%
Estimated time to completion    : 04:12:30
Number of Groups                : 3

Group 1:
{
Group Attributes              : Optimizer Generated
Group State                   : InProgress
Time Started                  : Tue Dec 22 21:30:23 2009
Time Completed                : N/A
Percent Complete              : 20%
Estimated time to completion : 00:40:12

Swap Pairs (2)
{
Source Device                           Target Device
-----------------------------------  -----------------------------------
Dsk                                     Dsk
Sym   Grp Group Name        Prot      Sym   Grp Group Name        Prot
----  --- ----------------  --------  ----  --- ----------------  --------
0023  004 sata_disks        R5(3+1)   0088  004 sata_disks        R5(3+1)
0032  004 sata_disks        R1        0058  004 sata_disks        R1
}
}

Group 2:
{
Group Attributes              : FAST Generated(Performance)
Group State                   : NotStarted
Time Started                  : Tue Dec 22 20:30:23 2009
```

```
Time Completed               : N/A
Percent Complete             : 0%
Estimated time to completion : 01:35:31


Swap Pairs (2)
{
Source Device                          Target Device
------------------------------------- -------------------------------------
Storage                                Storage
Sym  Tier Name   Prot     Group Name  Sym  Tier Name   Prot     Group Name
---- ----------- -------- ----------- ---- ----------- -------- -----------
0045 PrimeTier   R1       OraSales    00E0 PrimeDbTier R5(3+1)  OraSales
0046 PrimeTier   R1       OraSales    00E1 PrimeDbTier R5(3+1)  OraSales
}
}

Group 3:
{
Group Attributes             : FAST Generated(Compilance)
Group State                  : NotStarted
Time Started                 : Tue Dec 22 21:30:23 2009
Time Completed               : N/A
Percent Complete             : 0%
Estimated time to completion : 04:12:30

Move Devices (6)
{
Source Device                       Target
--------------------- ---------------------------------------------
Storage                                   Dsk
Sym  Group Name       Tier Name        Prot     Grp Group Name
---- ---------------- ---------------- -------- --- ----------------
0042 OraSales         ArchiveTier      R5(7+1)  004 sata_disks
0043 OraSales         ArchiveTier      R5(7+1)  004 sata_disks
0044 OraSales         ArchiveTier      R5(7+1)  004 sata_disks
0059 OraSales         WorkDbTier       R5(3+1)  001 fiber_disk
005A OraSales         WorkDbTier       R5(3+1)  001 fiber_disk
005B OraSales         WorkDbTier       R5(3+1)  001 fiber_disk
}
}
```

Note: FAST-generated plans will also display in the Optimizer output. There are no plans for thin tiers.

Approving a swap list

User approval mode allows you to see which swaps will occur before they take place. To set Optimizer to user approval mode, use the `set control_parms swap_mode=USER_OK;` line in the command file. Once this is set, Optimizer recalculates a swap list approximately once every hour as long as samples have been collected for the specified minimum performance period. At any time, the latest swap list can be retrieved using the `symoptmz show -swap_list` command. The swap list can then be approved or declined using the following syntax in the command file:

```
set swap <APPROVE | DECLINE>
  [begin_at=TimeVal,]
  TIMESTAMP=TimeVal;
```

The time stamp specified must be the time stamp returned by the last `symoptmz show -swap_list` command. If the command to approve or decline the swap list returns an error, the swap list is probably out of date and a new one is available from the service processor. The latest swap list should then be retrieved again with the `symoptmz show -swap_list` command.

"Reviewing Optimizer plans" on page 527 provides examples of output from the `symoptmz show -swap_list` command.

Performing rollbacks

Rollbacks are used to undo swaps that conflict with your business rules. Rollbacks are performed by first showing the rollback list, as follows:

```
symoptmz show -rollback_list
```

Use the following command file syntax to rollback a swap:

```
set swap <APPROVE | DECLINE>
  [begin_at=TimeVal,]
  TIMESTAMP=TimeVal,
  [,ROLLBACK];
```

The rollback feature is an *all or nothing* feature. Optimizer reverses all swaps by going backward from the present to a selected earlier time and undoing each and every swap. Use the User Approved mode as described in "Approving a swap list" on page 533 to undo specific swaps. Rollback swaps can be scheduled for execution according to normal Optimizer policy.

Note: When Optimizer rolls back a swap, if FAST swaps are in the swap list, they will be affected. For more information about FAST, see "Fully Automated Storage Tiering (Enginuity 5874 to 5876)" on page 166.

Reading Optimizer logs

To retrieve the Optimizer activity log, including swaps that have been completed, use the following command:

```
symoptmz read -log_type RUNTIME [-start DateTime] [-stop DateTime]
```

To retrieve the Optimizer error log, use the command:

```
symoptmz read -log_type ERROR [-start DateTime] [-stop DateTime]
```

The log that is returned includes DOS line-feed characters that can be removed using sed or another text editor:

```
cat opt_log.txt | sed s/^M$// > cleaned_opt_log.txt
```

In a UNIX shell, the ^M should be typed using **CTRL-V CTRL-M**.

In addition, the `symoptmz read` command can be used to determine why a swap was not performed.

Migrating devices

Note: The symoptmz migrate command is not available on arrays running Enginuity 5874 and higher. For those arrays, use the symmigrate command, as explained in Chapter 13, "Virtual LUN Technology (Enginuity 5874 through 5876)."

When devices are moved to a new set of disks, this is called a *migration.* A migration can be used to populate newly added disk drives, or to move devices between high performance and high capacity disks. A migration can only be initiated by a user. When devices are moved to a new location, this is called a *relocation*, and it may be used to balance disk usage or to fill in holes left by deleting devices. Relocations are performed by the Optimizer automatically, as needed. A relocation cannot be initiated by a user.

The Optimizer `symoptmz` command provides the following features for data migrations:

- The source devices to be migrated can be specified as a device group.
- The target disks can be a disk group number.
- You can optionally specify that unmapped and/or unmasked devices can be used as target devices.

When devices or device group devices are migrated to target disks or disk groups, Optimizer looks for free space on the target disks for the migration. If the target disks do not have any free space, you can specify to use unmapped and/or unmasked devices. These options indicate that source devices can be migrated to space occupied by unmapped and/or unmasked devices on the target disks. When this option is specified, Optimizer will NOT look for free space on the target disks but will only look for matching unmapped and/or unmasked devices.

Note: The unmapped and unmasked options are not recommended if there is free space on the target disks.

As with all device migrations, devices on the target disk list must match the size, emulation, and protection type of the source devices for the migration to succeed. To initiate a migration, use the `symoptmz` command with the following syntax in the command file:

```
migrate
   dev[s] SymDevStart1[:SymDevEnd1]
   [,SymDevStart2[:SymDevEnd2],...]
   TO disk[s] {diskn} [,{diskn},...]
   [unmapped=TRUE] [unmasked=TRUE]
   [begin_at=TimeVal];

migrate
   device_group DgName
   TO disk_group_num #
   [unmapped=TRUE] [unmasked=TRUE]
   [begin_at=TimeVal];

migrate
   device_group DgName
   TO disk[s] {diskn} [,{diskn},...]
   [unmapped=TRUE] [unmasked=TRUE]
   [begin_at=TimeVal];
```

```
migrate
   dev[s] SymDevStart1[:SymDevEnd1]
   [,SymDevStart2[:SymDevEnd2],...]
   TO disk_group_num #
   [unmapped=TRUE] [unmasked=TRUE]
   [begin_at=TimeVal];
```

Where:

The `diskn` is of the form `DDD,I,T`:

`DDD` is the director Identifier,

`I` is the Director Interface, and

`T` is the Target ID

The *`TimeVal`* is in the form of `MMDDYYYY:HHMMSS`.

Example An example of a device migration command file follows:

```
migrate devices 240, 245:247 to disks {16B, C, 0}, {4D, C, 0};
migrate device_group Mydgname to disk_group_num 1;
migrate devices 240, 245:247 to disk_group_num 2;
migrate device_group Mydgname to disks {16B, C, 0}, {4D, C, 0};
migrate devices 240, 245:247 to disks {16B, C, 0}, {4D, C, 0} unmapped=TRUE;
migrate device_group Mydgname to disk_group_num 0 unmasked=TRUE;
migrate devices 240, 245:247 to disk_group_num 0 unmapped=TRUE unmasked=TRUE;
```

CHAPTER 18
Managing Time Windows (Enginuity 5876 or lower)

This chapter explains how to add and remove time windows using the `symtw` command. Instructions for converting time windows created in earlier versions of Solutions Enabler are also included.

- Overview 538
- Converting time windows 539
- Adding time windows 541
- Removing time windows 542
- Displaying time windows 543

Overview

IMPORTANT

Time windows are not supported on arrays running HYPERMAX OS 5977 or higher.

Time windows are used by FAST, FAST VP, and Symmetrix Optimizer to collect performance statistics and execute data movement within the array. Use the `symtw` command to manage time windows. This command improves on the process and formatting of time windows (previously managed with `symoptmz set time_window`).

You can continue to use the `symoptmz` command, or you can convert previously defined time windows to the new format using the `symtw convert` command. However, you cannot use both commands. In addition, the `symtw` command is not available until a `symtw convert` operation executes on a VMAX or DMX array.

Note: After using the `symtw` command, the `symoptmz` command for setting time windows is no longer supported.

There are three different types of time windows:

- Data movement for disk group provisioned devices (`-dp`).
- Data movement for virtually provisioned devices (`-vp`).
- Performance time windows which control the collection of statistics.

In addition, a defined time window needs to be specified as either *inclusive*, which allows the operation to be executed repetitively, or *exclusive* which prevents the operation for a future specific date and time.

The new time window definition includes the following features:

- All inclusive time windows are similar to the weekly by day time window definitions without the start and end date.
- The inclusive time windows are defined by using one or more days of the week and the start/end time to be applied to each day. The start and end time are in 30 minute increments from 00:00 to 24:00. The time 00:00 represents midnight AM and 24:00 represents midnight PM.
- The exclusive time windows can be defined for a period of time with the start date/time and end date/time. The start and end time are in 30 minutes increments from 00:00 to 24:00. The exclusive time window will have the highest priority and it will override any time windows that have been defined during those time period.
- The time window definitions stored on the array database are in GMT time. There is an option in the API to display the host local time when adding, removing, or querying the time windows.
- Any newly added time windows will not replace the current time windows. They will be added on top of the current time windows. The remove operation will allow the user to remove any specified time windows.
- Any expired exclusive time windows will be deleted whenever the time window database is updated.

- The `symtw` command supports a `convert` command to translate any legacy time window definitions defined previously using the `symoptmz` command to the enhanced `symtw` time window definitions. After the execution of the `symtw convert` command, time window definition and reporting using the `symoptmz` command will be blocked and all time window management and reporting must be performed using the `symtw` command.
- There are three system default time windows and the FAST VP controller will collect performance statistics and execute data movement only after a `symtw convert` operation by the user.
- VMAX 10K/VMAXe arrays only support `symtw` for defining FAST VP time windows. There are predefined performance time windows. They can optionally be removed using the `symtw remove` or `symtw rmall` commands.

Time window restrictions and guidelines

The following guidelines apply to the `symtw` type time window:

- For the inclusive time windows, the start time and end time cannot be extended to the previous or next day. Any time windows that extend to the next day need to be defined by using a separate `symtw` add commands with the same type.
- For the exclusive time windows, the time window start date/time and end date/time must be specified and the dates must in the future time.
- For both inclusive and exclusive time windows, the valid start and end time values for the days are from 00:00 to 24:00 in 30 minutes increments. The time 00:00 represents midnight AM and 24:00 represents midnight PM.

Time window weekly format

Inclusive time windows *always* utilize a Sunday to Saturday week. Therefore, if you enter a command, such as `symtw convert` with Wednesday's date, the system begins the conversion process beginning with the previous Sunday and through the seven days to the next Saturday.

Any time windows created in the previous week are deleted, and any time windows in the future after Saturday are deleted. The only exception is a legacy 'exclude' window for one time in the future.

Converting time windows

The `symtw convert` command converts any legacy time window definitions defined through the `symoptmz` command to the enhanced `symtw` time window format.

You have the option to choose the start date when converting the time windows. When the date is specified in the command line, the enhanced time window definitions are created based on the composite time windows generated from the 7 days starting from the specified date. The converting date will be the current day if the user does not provide the exact date information.

Conversion guidelines

Follow these guidelines when converting legacy time windows:

- You must execute the `convert` command before you can use the `symtw` command to manage and report on time windows. All `symtw` commands will return an error if the legacy time window definition format still exists on the storage array.
- When the `symoptmz` time window definitions cannot be translated to the new `symtw` time window definitions, an error returns. The `-force` option can be specified to force the best effort conversion of the time windows.
- The `symoptmz` command is not supported after a successful `symtw convert` operation creates the enhanced time window format on the storage array.
- The `symtw` command is not supported if the time window database has been initialized and the format is in the legacy `symoptmz` time window format. When the time window database is uninitialized, the `convert` command is also required and it will set the database to the enhanced time window format.
- Only the time windows defined during the 7 days including the start date will be converted. All inclusive time windows defined before or after this period of time will be deleted.
- Exclusive time windows previously defined for a future date will convert to the enhanced time window format only if it is defined as `ONCE`.
- The start and end time of the time windows will be rounded to the nearest half hour time. The start time will be rounded down to the previous half hour and the end time will be rounded up to the next half hour.

Conversion examples

The following command will convert the inclusion legacy time window definitions, defined from Feb 6, 2011 through Feb 12, 2011, to the enhanced time window format. In addition all 'once' exclusion legacy time window definitions from Feb 6, 2011 and later will be converted to the enhanced time window format.

The following examples will display the output of `symtw convert` given the legacy time windows, assuming they are in file `legacy_windows` (the contents shown below), and the `symoptmz -sid 432 -f legacy_windows commit` command was executed:

```
set time_window id=MyPerf type=perf flag=include period=once
   starting=02202010:000000 ending=02202012:000000;
set time_window id=MyPerf1 type=perf flag=exclude period=WEEKLY
   starting=11042010:200000 ending=12302010:210000 days=WED_START FRI
   start_time=19:00 end_time=20:00;
set time_window id=MyDpDataMove type=swap provisioning=dp flag=include
   period=WEEKLY_BY_DAY starting=11042010:200000 ending=12302012:210000 days=SUN
   MON TUE THUR start_time=19:00 end_time=2:00;
set time_window id=MyDpDataMoveExclude type=swap provisioning=dp flag=exclude
   period=ONCE starting=02112011:120000 ending=02112011:131500;
set time_window id=MyDpDataMoveExclude1 type=swap provisioning=dp flag=exclude
   period=ONCE starting=11242010:200000 ending=11252010:210000;
set time_window id=MyDpDataMoveExclude2 type=swap provisioning=dp flag=exclude
   period=ONCE starting=02142011:200000 ending=02142011:210000;
set time_window id=MyVpDataMove type=swap provisioning=vp flag=include
   period=WEEKLY starting=11042010:200000 ending=12302012:210000 days=WED_START FRI
   start_time=19:00 end_time=20:00;
```

```
set time_window id=MyVpDataMove2 type=swap provisioning=vp flag=include
   period=WEEKLY starting=02112011:200000 ending=12302012:210000 days=WED_START FRI
   start_time=19:00 end_time=20:00;
set time_window id=MyVpDataMoveExclude type=swap provisioning=vp flag=exclude
   period=WEEKLY starting=03042011:200000 ending=12302011:210000 days=WED_START
   FRI;
```

In this conversion example the `-force` option is added for a “best effort” conversion:

```
symtw convert  -sid 432 -force

Execute conversion of legacy time windows for Symmetrix 000000000432

Evaluating DP Move Time Window 'The Default Time Window': Convertible
Evaluating VP Move Time Window 'The Default Time Window': Convertible
Evaluating Performance Time Window 'The Default Time Window': Convertible
Evaluating Performance Time Window 'MyPerf ': Convertible
Evaluating Performance Time Window 'MyPerf1': Expired
Evaluating DP Move Time Window 'MyDpDataMove': Convertible
Evaluating DP Move Time Window 'MyDpDataMoveExclude': Convertible
Evaluating DP Move Time Window 'MyDpDataMoveExclude1': Expired
Evaluating DP Move Time Window 'MyDpDataMoveExclude2': Convertible
Evaluating VP Move Time Window 'MyVpDataMove': Convertible
Evaluating VP Move Time Window 'MyVpDataMove2': Partially Convertible - End of
   period exceeds enhanced time window end date
Evaluating VP Move Time Window 'MyVpDataMoveExclude': Not-Convertible - Weekly
   exclude starts after enhanced time window end date.


Converting DP Move Time Window 'The Default Time Window': Done
Converting VP Move Time Window 'The Default Time Window': Done
Converting Performance Time Window 'The Default Time Window': Done
Converting Performance Time Window 'MyPerf ': Done
Converting Performance Time Window 'MyPerf1': Skipped
Converting DP Move Time Window 'MyDpDataMove': Done
Converting DP Move Time Window 'MyDpDataMoveExclude': Done
Converting DP Move Time Window 'MyDpDataMoveExclude1': Skipped
Converting DP Move Time Window 'MyDpDataMoveExclude2': Done
Converting VP Move Time Window 'MyVpDataMove': Done
Converting VP Move Time Window 'MyVpDataMove2': Done
Converting VP Move Time Window 'MyVpDataMoveExclude': Skipped

Conversion of legacy Time Windows successfully completed.
```

Adding time windows

Use the following syntax to create a time window:

```
symtw -sid SymmID -inclusive [-noprompt]
          -type <move_dp | move_vp | perf | all>
    add   -days DayList -start_time Time -end_time Time

symtw -sid SymmID -exclusive [-noprompt]
          -type <move_dp | move_vp | perf | all>
    add   -start_day DateTime -end_day DateTime
```

Where:

`-inclusive` — Specifies that you want to create this time window.

`-exclusive` — Specifies that you want the system to exclude these times from any time window.

`-noprompt` — Requests that no prompts are returned after the command is entered. The default is to prompt the user for confirmation.

`-type` — Indicates the type of time window, as follows:

- `move_dp` — Move data on disk provisioned devices.
- `move_vp` — Move data on virtually provisioned devices.
- `perf` — Gather performance data.
- `all` — Create all three types of time windows.

`Daylist` — Specifies the days for the time window. Possible values are: Any comma-separated combination of `MON`, `TUE`, `WED`, `THU`, `FRI`, `SAT`, and `SUN`.

`Time` — Specifies the time of day in the format of `HH:MM`. Possible values are `00:00` to `24:00` for each day in 30 minute increments. The time `00:00` represents midnight AM, and `24:00` represents midnight PM.

`DateTime` — Specifies the date and time in the format of `MMDDYYYY:HHMM`. The time of day is in 30 minute increments. The valid values for minutes are `0` and `30`.

Note: Inclusive time windows only use the *`DayList`* and *`Time`* parameters. Exclusive time windows only use the *`DateTime`* parameter.

In the following examples, two *inclusive* time windows are added:

```
symtw -sid 397 add -type MOVE_DP -inclusive
     -days Mon,Tue,Wed,Thu,Fri
     -start_time 18:00 -end_time 24:00

symtw -sid 397 add -type MOVE_VP -inclusive
     -days Mon,Tue,Wed,Thu,Fri
     -start_time 08:00 -end_time 17:30
```

In the following examples, two *exclusive* time window are added:

```
symtw -sid 397 add -type MOVE_DP -exclusive
     -start_day 11042011:0000 -end_day 11042011:2300

symtw -sid 397 add -type MOVE_DP -exclusive
     -start_day 12252011:0000 -end_day 12262011:2400
```

Removing time windows

Use the `symtw remove` command to execute the following operations:

- Remove the specified inclusive time windows using the `-type`, `-days`, `-start_time` and `-end_time` options.
- Remove the specified exclusive time window with `-type`, `-start_date` and `-end_date` options.

This command does not require an exact match between the existing time windows and the input time windows from the command when removing the time windows.

Use the following syntax for removing time windows:

```
symtw -sid SymmID -inclusive [-noprompt]
           -type <move_dp | move_vp | perf | all>

    remove -days DayList -start_time Time -end_time Time
```

```
symtw -sid SymmID -exclusive [-noprompt]
           -type <move_dp | move_vp | perf | all>

    remove -start_day DateTime -end_day DateTime
```

These options are defined in "Adding time windows" on page 541.

The following example removes the time windows from 17:30 to 18:30 on Monday, Tuesday, Wednesday, Thursday and Friday. In this example, the existing time windows are defined as Monday to Friday and the starting and ending time are from 18:00 to 24:00 for each day.

```
symtw -sid 397 remove -type MOVE_DP -inclusive
    -days Mon,Tue,Wed,Thu,Fri
   -start_time 17:30 -end_time 18:30
```

After the execution of the `remove` command, the new time windows will be Monday to Friday and the starting and ending times are from 18:30 to 24:00 for each day.

The following example removes the exclusive time window:

```
symtw -sid 397 remove -type MOVE_DP -exclusive
  -start_day 11042011:0000 -end_day 11042011:0800
```

Removing all time windows

The `symtw rmall` command allows you to clear time windows, as follows:

- Remove all inclusive time windows of one type or more with the `-type` and `-inclusive` options.
- Remove all exclusive time windows of one type or more with the `-type` and `-exclusive` options.
- Remove all exclusive and inclusive time windows of one type or more with the `-type`, `-exclusive`, and `-inclusive` options.

For example, the following command removes all time window definitions:

```
symtw -sid 397 rmall -type ALL -exclusive -inclusive
```

Displaying time windows

To display the defined time windows, use the `symtw list` command. Use the `-type` option to filter the display for only the requested window types. The list command provides the following options:

```
symtw [-sid SymmID] [-offline]

   list [-type <move_dp | move_vp | perf>]
   list -summary [-date Date]
```

Where:

`-type` — Displays only the specified type.

`-summary` — Displays a calendar view of all defined time windows.

`-date` — Displays the defined time window for the specified date. The date format is: `MMDDYYYY`

The following example shows the time windows defined for array 397:

```
symtw -sid 397 list

Symmetrix ID: 000194900397

  DP Data Movement Time Windows

    Sunday       : None
    Monday       : 18:30 - 24:00
    Tuesday      : 18:30 - 24:00
    Wednesday    : 18:30 - 24:00
    Thursday     : 18:30 - 24:00
    Friday       : 18:30 - 24:00
    Saturday     : None

    Exclusive Time Windows (2)
      {
        Fri Nov  4 08:00:00 2011 - Fri Nov  4 23:00:00 2011
        Fri Dec 25 00:00:00 2011 - Sat Dec 26 24:00:00 2011
      }

  VP Data Movement Time Windows

    Sunday       : None
    Monday       : 08:00 - 17:30
    Tuesday      : 08:00 - 17:30
    Wednesday    : 08:00 - 17:30
    Thursday     : 08:00 - 17:30
    Friday       : 08:00 - 17:30
    Saturday     : None

    Exclusive Time Windows (0)

  Performance Time Windows

    Sunday       : 07:00 - 18:00
    Monday       : 07:00 - 18:00
    Tuesday      : 07:00 - 18:00
    Wednesday    : 07:00 - 18:00
    Thursday     : 07:00 - 18:00
    Friday       : 07:00 - 18:00
    Saturday     : 07:00 - 18:00

    Exclusive Time Windows (0)
```

Using the -summary option

Use the `symtw list -summary` command to show a summary of the time window definitions for the current week starting from Sunday. Any exclusive time window that overrides the defined time windows during the current time of period also display.

```
symtw list -sid 397 -summary

Symmetrix ID: 0001949000397

                                Time Window Summary
                |-------------------------------------------------------|
                |  SUN  |  MON  |  TUE  |  WED  |  THU  |  FRI  |  SAT  |
                |-------|-------|-------|-------|-------|-------|-------|
                | D V P | D V P | D V P | D V P | D V P | D V P | D V P |
                |-------|-------|-------|-------|-------|-------|-------|
00:00 - 00:30   | . . . | . . . | . . . | . . . | . . . | . . . | . . . |
00:30 - 01:00   | . . . | . . . | . . . | . . . | . . . | . . . | . . . |
01:00 - 01:30   | . . . | . . . | . . . | . . . | . . . | . . . | . . . |
```

```
01:30 - 02:00 | . . . | . . . | . . . | . . . | . . . | . . . | . . . |
02:00 - 02:30 | . . . | . . . | . . . | . . . | . . . | . . . | . . . |
02:30 - 03:00 | . . . | . . . | . . . | . . . | . . . | . . . | . . . |
03:00 - 03:30 | . . . | . . . | . . . | . . . | . . . | . . . | . . . |
03:30 - 04:00 | . . . | . . . | . . . | . . . | . . . | . . . | . . . |
04:00 - 04:30 | . . . | . . . | . . . | . . . | . . . | . . . | . . . |
04:30 - 05:00 | . . . | . . . | . . . | . . . | . . . | . . . | . . . |
05:00 - 05:30 | . . . | . . . | . . . | . . . | . . . | . . . | . . . |
05:30 - 06:00 | . . . | . . . | . . . | . . . | . . . | . . . | . . . |
06:00 - 06:30 | . . . | . . . | . . . | . . . | . . . | . . . | . . . |
06:30 - 07:00 | . . . | . . . | . . . | . . . | . . . | . . . | . . . |
07:00 - 07:30 | . . P | . . P | . . P | . . P | . . P | . . P | . . P |
07:30 - 08:00 | . . P | . . P | . . P | . . P | . . P | . . P | . . P |
08:00 - 08:30 | . . P | . V P | . V P | . V P | . V P | . V P | . . P |
08:30 - 09:00 | . . P | . V P | . V P | . V P | . V P | . V P | . . P |
09:00 - 09:30 | . . P | . V P | . V P | . V P | . V P | . V P | . . P |
09:30 - 10:00 | . . P | . V P | . V P | . V P | . V P ->. V P<- . V P | . . P |
10:00 - 10:30 | . . P | . V P | . V P | . V P | . V P | . V P | . . P |
10:30 - 11:00 | . . P | . V P | . V P | . V P | . V P | . V P | . . P |
11:00 - 11:30 | . . P | . V P | . V P | . V P | . V P | . V P | . . P |
11:30 - 12:00 | . . P | . V P | . V P | . V P | . V P | . V P | . . P |
12:00 - 12:30 | . . P | . V P | . V P | . V P | . V P | . V P | . . P |
12:30 - 13:00 | . . P | . V P | . V P | . V P | . V P | . V P | . . P |
13:00 - 13:30 | . . P | . V P | . V P | . V P | . V P | . V P | . . P |
13:30 - 14:00 | . . P | . V P | . V P | . V P | . V P | . V P | . . P |
14:00 - 14:30 | . . P | . V P | . V P | . V P | . V P | . V P | . . P |
14:30 - 15:00 | . . P | . V P | . V P | . V P | . V P | . V P | . . P |
15:00 - 15:30 | . . P | . V P | . V P | . V P | . V P | . V P | . . P |
15:30 - 16:00 | . . P | . V P | . V P | . V P | . V P | . V P | . . P |
16:00 - 16:30 | . . P | . V P | . V P | . V P | . V P | . V P | . . P |
16:30 - 17:00 | . . P | . V P | . V P | . V P | . V P | . V P | . . P |
17:00 - 17:30 | . . P | . V P | . V P | . V P | . V P | . V P | . . P |
17:30 - 18:00 | . . P | . . P | . . P | . . P | . . . | . . P | . . P |
18:00 - 18:30 | . . . | . . . | . . . | . . . | . . . | . . . | . . . |
18:30 - 19:00 | . . . | D . . | D . . | D . . | D . . | E . . | . . . |
19:00 - 19:30 | . . . | D . . | D . . | D . . | D . . | E . . | . . . |
19:30 - 20:00 | . . . | D . . | D . . | D . . | D . . | E . . | . . . |
20:00 - 20:30 | . . . | D . . | D . . | D . . | D . . | E . . | . . . |
20:30 - 21:00 | . . . | D . . | D . . | D . . | D . . | E . . | . . . |
21:00 - 21:30 | . . . | D . . | D . . | D . . | D . . | E . . | . . . |
21:30 - 22:00 | . . . | D . . | D . . | D . . | D . . | E . . | . . . |
22:00 - 22:30 | . . . | D . . | D . . | D . . | D . . | E . . | . . . |
22:30 - 23:00 | . . . | D . . | D . . | D . . | D . . | E . . | . . . |
23:00 - 23:30 | . . . | D . . | D . . | D . . | D . . | D . . | . . . |
23:30 - 24:00 | . . . | D . . | D . . | D . . | D . . | D . . | . . . |
              |-------|-------|-------|-------|-------|-------|-------|

Legend: D = Disk Group Provisioning Movement Time Window
        V = Virtual Provisioning Movement Time Window
        P = Performance Time Window
        E = Time Windows Overridden by the Exclusive Time Windows
```

CHAPTER 19
Performing Double Checksum Operations (Enginuity 5876 or lower)

This chapter describes Double Checksum concepts and how to minimize the impact of I/O errors on database consistency during I/O transfers between hosts and array storage devices using the Double Checksum SYMCLI commands.

- Overview 548
- Implementing EMC Double Checksum for Oracle 549
- Implementing Generic SafeWrite for generic applications 553
- Syntax and examples 557

Overview

IMPORTANT

Double Checksum is not supported on arrays running HYPERMAX OS 5977 or higher.

The Double Checksum feature provides a method to help minimize the impact of I/O errors on database consistency during I/O transfers between hosts and VMAX or DMX array storage devices.

For Oracle, **EMC Double Checksum for Oracle** contains a rich set of checks that can be natively performed by the array. For each Relational Database Management System (RDBMS) write in the array, checksum values are computed and compared to test the data for any corruption picked up along the way from the host. Although errors of this kind are infrequent, they can have a considerable effect on data availability and recovery. Refer to "Implementing EMC Double Checksum for Oracle" on page 549 for details on this feature.

For generic RDBMS applications, **EMC Double Checksum for Generic Applications** provides the Generic SafeWrite[1] feature to help protect critical applications from incurring an incomplete write, and subsequent torn page, due to a failure with a component connected to the array Front End Channel Adapter. Generic SafeWrite is most often used to protect against corruption are from HBA and link failures including server crashes, where essentially, it will help protect against fractured writes that can occur before the data reaches the array. Refer to "Implementing Generic SafeWrite for generic applications" on page 553 for details on this feature.

This chapter contains overview and concept information; see the `symchksum` command in the *EMC Solutions Enabler SYMCLI Command Reference* for the complete list of syntax and options.

Traditional methods of preventing data corruption

Data corruption checking is an integral part of most RDBMS products. For instance, Oracle computes a checksum that verifies the data within each page. If corruption occurs, the checksum will be incorrect when the data is read. However, this checking only takes place within the host system—not the storage array.

Although data appears to the host to travel directly to the array, it passes through multiple hardware and software layers. These can lead to problems such as corruption introduced by errors in the operating system or the I/O driver. Hardware can also introduce corruption. This means that valid data within the RDBMS might arrive corrupted at the storage device. The storage device writes the data as is, because it has no way of validating the data.

As a result, if there is corruption after the data has left the host system, it will not be detected until that data is read back into the system, which can be much later. The RDBMS issues an alert, and then the data must be rebuilt from backups and database logs. While a corruption remains undetected, the number of logs required for recovery increases. This causes the data recovery process to be more complex and time-consuming.

1. Generic SafeWrite has been created with RDBMS in mind. Applications intended for use with Generic SafeWrite include, but are not limited to Microsoft Exchange, Microsoft SQL Server, DB2/UDB and Oracle. Generic SafeWrite is not intended for use with applications where torn pages are not a concern, such as fileshares or FTP servers.

Benefits of checking within the array

With this feature, the array can perform error checks on data pages handled within a checksum extent *as they are written to the disk.* The check takes place before the write command is acknowledged. If an error is detected within the blocks of the extent, the I/O can be rejected or reported in a phone home connection or logged in the VMAX or DMX error log facilities. The error action can be specified by the administrator.

This checking feature minimizes the possibility of data corruption occurring between the host and the array. It improves the recovery time by flagging the error at the time of the "write." When this error condition is raised, and reject I/O is selected, Oracle takes an action, such as taking the tablespace offline.

Implementing EMC Double Checksum for Oracle

The `symchksum` command allows you to perform control operations that manage the checksum I/O extents.

For example, to enable Oracle checksum checking on the extents of all the devices that define the current database instance and then to phone home on error, enter:

```
symchksum enable -type Oracle -phone_home
```

This command requires an Oracle PAK license.

The following are current restrictions for this feature:

- Oracle databases version 8.0 and above are supported.
- Data block size is limited to 32 KB.
- Checksum data objects can only be Oracle data files, control files, and redo logs.
- Devices cannot be diskless devices (DLDEVs).

The Oracle instance being checked must be started with the following `init.ora` configuration parameter set to true: db_block_checksum=true.

Note: Manipulating remote devices over an SRDF link is not supported. The `symchksum` command can only manipulate devices in locally attached arrays.

Other checksum operations

The following additional checks are available:

- MagicNumber — Verifies the 3-bit magic number that appears in Oracle data blocks. (Enabled by default.)
- NonzeroDba — Checks for non-zero data block address. The dba stored in a data block is never zero. (Enabled by default.)
- Check_All_Blocks — Applies checks to each block in a write. Otherwise, only the first block is checked.
- Straddle — Checks that the write does not straddle known Oracle areas.

- Check_dba — Checks that the logical address embedded by Oracle is compatible with the storage address of the writes.

Note: If `-check_dba` is selected, extents from separate datafiles cannot be collapsed. The `-check_dba` option requires data specific to each datafile, which would be lost in an extent collapse.
As a result, each datafile will occupy at least one array device extent. When using a logical volume manager and `-check_dba`, this creates a limitation of 200 datafiles per volume group or disk group.

To enable Oracle checksum and add the optional operations of check DBA and check all blocks, enter:

```
symchksum -type Oracle enable -check_dba -check_all_blocks
```

With the addition of these new tests, the output when listing the extents will look similar to the following:

```
Symmetrix ID: 000187900671

          D E V I C E S   W I T H   C H E C K S U M   E X T E N T S

                                                     Action  Checks

                                                       R P  C   C   A N   D
                                                       e h  h M h S l Z F i
                                                       j o  k a k t l   r s
                                                     L e n  s g D r B D a c
                                     Num Blk         o c e  u i B a l B c r
Device Name                        Dev Exts Siz Type  g t H  m c A d k A t d
-------------------------------------------------------------------------------

/dev/sdj                          0048    1 16b Oracle  X X X  X X . . . X X X
/dev/sdz                          0058    1 16b Oracle  X X X  X X . . . X X X

2 Devices with Checksum extents were found.
```

You can use this output to determine which features are enabled on the devices with checksum extents.

Suppressing default features

By default, the `symchksum` command attempts to select the following tests:

- Block checksum value (Checksum)
- Blocksize verification (BlkSize)
- 3-bit magic number (MagicNumber) in Oracle data blocks (not available for redo logs)
- Checks for non-zero data block addresses (NonZeroDba)

To turn off any of the automatic checksum features, use the `-suppress_feature` option and supply the name of the feature, for example:

```
symchksum -type Oracle enable -suppress_feature MagicNumber
```

The `-suppress_feature` option is only for the operations that run by default. If you want to turn off an option that was manually enabled, such as `-phone_home`, you must disable the checksum operation and begin again with a new `symchksum enable` command.

Enabling checksum options

When using the `symchksum enable` command, you can decide if you would like to reject the I/O, or have the array phone home when a checksum error is detected.

If an I/O is not a multiple of the object block size, you can choose to reject the I/O. This is called a fractured I/O, and is selected with the `-fractured_reject_io` option. When using this option, the `-reject_io` option must also be used.

When extents are enabled with the `-discard` option, EMC Double Checksum writes blocks to disk until a failed block is detected. The `-discard` option divides a large I/O into smaller units of 32 KB each. When a checksum failure is detected, all blocks in that unit and subsequent units are discarded. When using this option, the `-reject_io` option must also be used.

The `symchksum enable` command understands the Oracle database structure. The feature can be enabled for tablespaces, control files, redo logs, or the entire database.

Note: For Oracle 9i and above, if the block size for a tablespace is altered, then the user must disable and then re-enable the extents of the tablespace to ensure that the block size of the enabled extents match the block size of the tablespace.

Note: When FF or power down occurs, extents are lost. You must run the `symchksum enable` command again.

Verifying checksum is enabled

The `symchksum` command also allows you to verify that the data file's extents are currently enabled for checksum checking. This provides an easy way to determine if the specified tablespace or instance is fully protected by the storage array checksum feature. The `verify` action will report if all, some, or none of the Oracle data file's extents are enabled for checksum checking. This is useful in environments where the database configuration changes frequently. An example of this is if a new data file was added but checksum was not enabled for the new file.

The `symchksum verify` command understands the Oracle database structure. The feature can be enabled for tablespaces, control files, redo logs, or the entire database.

Validating for checksum operations

The `symchksum` command also allows you to validate your Oracle tablespace or instance for checksum operations without performing any active actions. This is helpful when you want to know if your database environment is configured to support VMAX or DMX checksum functionality without actually making any changes.

If the validate is successful, you can enable EMC Double Checksum on the specified Oracle database or tablespace. The following items are validated:

- Oracle version 8 or higher is installed

- Oracle's checksum initialization parameter is set (`db_block_checksum`)
- If the Oracle data file is created on a striped LVM, that the LVM stripe width is a multiple of the Oracle block size
- Oracle data file's block size is less than or equal to 32 KB
- The Enginuity version supports the checksum functionality
- Each DMX device (prior to Enginuity 5773) has the Checksum flag set
- Each VMAX or DMX device has a supportable number of extents defined

The `symchksum validate` command understands the Oracle database structure. The feature can be enabled for tablespaces, control files, redo logs, or the entire database.

Disabling checksum

The `symchksum disable` command understands the Oracle database structure. The feature can be enabled for tablespaces, control files, redo logs, or the entire database.

The `symchksum disable` command can also be used on a device basis. This capability is not normally used, but is provided in the event the tablespace was dropped before EMC Double Checksum was disabled for that object.

When the `disable` action is specified for a storage array device, the `-force` flag is required. Disabling extents in this way can cause a mapped tablespace or database to be only partially protected, therefore, use this option with caution. All the extents being monitored for checksum errors on the specified storage array device will be disabled.

Functionality

In support of the Oracle and EMC Hardware Assisted Resilient Data (HARD) initiative, EMC Double Checksum provides end-to-end database integrity by employing sophisticated error detection technology to detect and prevent corruption of data blocks written to VMAX or DMX arrays. Table 36 shows the **EMC Double Checksum for Oracle** functionality.

Table 36 EMC Double Checksum for Oracle functionality

Block checksum verification	Dial home alert	Discard	Max extents [a] / VMAX or DMX volume	Magic number detection	Zero dba detection	All blocks option [b]	Straddle [c]
Yes	Yes	Yes	200	Yes	Yes	Yes	Yes

a. VMAX or DMX array extents are internal to the array and are used to map Oracle tablespace locations; they are not the same as Oracle extents

b. For Magic Number Detection and Zero dba Detection, the default mode is to check only the first block in an I/O. If the All Blocks Option is selected in the SYMCLI, then Magic Number Detection and Zero dba Detection checks are performed for each Oracle block. For the checksum check, the only mode is to sum the I/O as a whole; it is not affected by the All Blocks Option.

c. Sd an extension to the Oracle and EMC HARD initiative, EMC Double Checksum functionality checks for I/O that overlaps defined database areas. For example, consider a database that has data from blocks 16 to 105. If an I/O is received for blocks 100 to 200, this I/O is said to straddle the database's data. This is unexpected and likely to be a potentially corrupting I/O. If the Straddle option is selected in the SYMCLI, then EMC Double Checksum will consider straddling I/O to be an error.

Implementing Generic SafeWrite for generic applications

Generic SafeWrite (GSW) is used to help protect critical applications from incurring an incomplete write, and subsequent torn page, due to a failure with a component connected to the storage array Front End Channel Adapter.

Using Generic SafeWrite to protect applications

A Relational Database Management System (RDBMS), such as Oracle and Microsoft SQL Server, structure data within database files using pages (also referred to as blocks). Pages within a database are the smallest allocation unit size possible for a database object (such as a table or a row).

For example, the page size for Microsoft SQL Server, and for Oracle, is 8 KB. Although this size is configurable, it is usually set to 8 KB. If an incomplete page is written to a database file, a corruption to the database will occur. The resulting corruption is commonly referred to as a *torn page*. Torn pages are only detected by most RDBMSs after the corruption has been written, when that area of the database is read, which could be long after when the corruption was introduced. In general, the only recovery from a torn page is to perform a restore from a backup (some RDBMSs allow page-level restores, while others require a complete database restore). Torn pages can occur due to failures in various components that lie between the RDBMS and the storage array.

The EMC Double Checksum Generic SafeWrite feature can help protect critical applications from incurring incomplete writes, and subsequent torn pages, due to a failure with a component connected to the storage array Front End Channel Adapter.

Most often, Generic SafeWrite will be used to protect against corruption that occurs when the HBA or link fails (including server crashes). In this scenario, Generic SafeWrite will protect against fractured writes that occur before the data reaches the array.

Generic SafeWrite is deemed *generic* because the checks performed to ensure complete data are application independent. For instance, Generic SafeWrite will not perform any Oracle- or SQL Server-specific checksums to verify data integrity. It is important to note that for Oracle, EMC Double Checksum for Oracle provides a rich set of checks which can be natively performed by the array. For more information on EMC Double Checksum for Oracle, refer to "Implementing EMC Double Checksum for Oracle" on page 549.

Where to enable Generic SafeWrite

Generic SafeWrite only needs to be enabled for specific devices on the VMAX or DMX array. For a RDBMS, Generic SafeWrite only needs to be enabled for devices that support data files. The list below gives an example of database files where the supporting devices for these files should have Generic SafeWrite enabled:

Microsoft Exchange:

- `.edb` files
- `.stm` files

Microsoft SQL Server:

- `.mdf` files
- `.ndf` files

Oracle:

- Data files
- Control files

Note: Manipulating remote devices over an SRDF link is not supported. The `symchksum` command can only manipulate devices in locally attached arrays.

It is recommended to enable Generic SafeWrite for database file devices, though it is not necessary to enable it for database log devices. In general, a RDBMS will write to its respective log file with a 512 byte sector alignment. The RDBMS can therefore determine the last sector that was correctly written and subsequently discard or rollback any transactions that are incomplete.

Note: It is always a best practice to separate the location of database files and log files for a given database onto unique devices. There are cases, however, where the data file and log file may share the same device. In this case, it is still possible to have GSW enabled, however, there will be a performance impact to the log writes which may impact application performance.

There are no restrictions regarding the size of a device or the number of devices where GSW can be enabled. However, for both Generic Safewrite and Oracle Safewrite only SRDF/S and TimeFinder/Mirror are supported. It is also supported to enable GSW on filesystems across all logical volume managers as well as on raw devices, given the OS platforms are supported by the Solutions Enabler Storage Resource Management (SRM) component. When using filesystems on Windows and Linux hosts, for performance reasons, it is strongly recommended to ensure the filesystems are properly aligned with the storage. For more information regarding filesystem alignment, refer to *Using diskpar and diskpart to Align Partitions on Windows Basic and Dynamic Disks* available on EMC online support.

Configuring Generic SafeWrite

To use Generic SafeWrite, you must:

- For Enginuity versions prior to 5773, enable the `RDB_cksum` device flag on all devices targeted for Generic SafeWrite
- Run a discover operation to update the physical device information in the SYMAPI database
- Enable Generic SafeWrite on all devices targeted for Generic SafeWrite use with the `symchksum` command

Setting the RDB_cksum device flag

Note: VMAX or DMX arrays do not require the `RDB_cksum` device flag enabled for Double Checksum and Generic SafeWrite use.

For Enginuity versions prior to 5773 only:

Before you can use Generic SafeWrite, the `RDB_cksum` array device flag must be enabled on all devices targeted for Generic SafeWrite use. This change does not turn Generic SafeWrite on, it only allows it to be enabled on the specified devices.

The `RDB_cksum` device flag can be set by using the SYMCLI `symconfigure` command, which will perform an array configuration change. For more information on using `symconfigure`, refer to Chapter 7, "Managing Configuration Changes."

Note: If `symconfigure` cannot be used, the appropriate device flag can also be set on the array by a EMC Customer Support Engineer.

The following is an example command:

```
symconfigure -sid 54 -f c:\enable_cksum.txt commit
```

Where the `c:\enable_cksum.txt` file contains the following command:

```
set device 00015:00019 attribute=RDB_Cksum;
```

Note: When using `symconfigure`, if metavolumes are used, this flag only the metahead needs to be specified.

Enabling Generic SafeWrite

Once the device flags are set on the array, it is then possible to use the `symchksum` command to enable Generic SafeWrite. Before running the `symchksum` command, confirm the following:

- The devices that you enabled for Generic SafeWrite are visible to the host from where the `symchksum` command will be run.
- Run a `symcfg discover` command after presenting devices to a host, in order to update the SYMAPI configuration database with the correct physical drive information.

Using the `symchksum` command, Generic SafeWrite can be enabled by specifying a specific device, a range of devices, or a device group.

Enabling for a device

To enable Generic SafeWrite for a device, use the command syntax shown in the example below:

```
symchksum enable -type generic dev 005 -sid 54
```

Note: If this is a metadevice, only the metahead needs to be specified.

Enabling for a range of devices

To enable Generic SafeWrite for a contiguous range of devices:

```
symchksum enable -type generic -devs 005:025 -sid 54
```

Enabling for a device group

To enable Generic SafeWrite for a device group:

```
symchksum enable -type generic -g sql_data -sid 54
```

Note: Enabling Generic SafeWrite on a Composite Group (CG) is currently not supported.

The `symchksum enable -type generic` command automatically sets the Log, Phone Home, and Generic Double Checksum options as described below:

- Log — Indicates that errors will be sent to the storage array error log. These events should be visible with the `symevent` command.
- Phone Home — Indicates that an error will initiate a call by the storage array to EMC Customer Service.
- Generic — Allows two functions to be performed by the VMAX or DMX array. First, when an incomplete write is detected, it will be rejected and the array will force the I/O to be retried from the host. Then, if the host is not available to retry the I/O, the write will be discarded, preventing it from being written to disk.

Disabling Generic SafeWrite

Generic SafeWrite can be disabled using the `symchksum disable -type generic` command as shown in the same examples below.

Disabling for a device

To disable Generic SafeWrite for a device, use the command syntax shown in the example below:

```
symchksum disable -type generic dev 005 -sid 54
```

Note: If this is a metadevice, only the metahead needs to be specified.

Disabling for a range of devices

To disable Generic SafeWrite for a contiguous range of devices:

```
symchksum disable -type generic -devs 005:025 -sid 54
```

Disabling for a device group

To disable Generic SafeWrite for a device group:

```
symchksum disable -type generic -g sql_data -sid 54
```

Listing Generic SafeWrite devices

To list which devices are Generic SafeWrite enabled, use the `symchksum list` command with the `-type generic` option. Only Generic SafeWrite-enabled devices that are visible to the host running the `symchksum list` command are returned. The `symchksum show` command can also be used to look at a specific device. For example:

```
symchksum show dev 103 -type generic -mb -sid 54
```

Performance considerations

Application performance should remain unaffected because database log writes, as well as database reads, will be performed normally. However, outside of application performance, there may be a slight increase in the write response time to the database file devices depending on application profile and usage. In general, this response time increase should not impact application performance. Writes to a database file are done asynchronously, therefore write response times to this file are less of a concern than to the log device.

Syntax and examples

This section contains the `symchksum` argument descriptions and several examples of using the SYMCLI `symchksum` command. See the *EMC Solutions Enabler SYMCLI Command Reference* for the complete list of syntax and options.

Table 37 symchksum syntax and descriptions

Command	Argument	Description
`symchksum`	`list`	Lists all devices that currently have checksum checking enabled. The list can be filtered by using the `-type` option (to show Oracle or generic).
	`show`	Displays the extents of a specified device that are having checksum checking performed.
	`enable`	Enables checksum checking on the extents of the specified devices.
	`disable`	Disables checksum checking on the extents of the specified devices.
	`validate`	Validates that a specified database or tablespace is able to have checksum checking enabled.
	`verify`	Verifies whether the specified database or tablespace has checksum checking enabled on all their devices.

To list the devices on array 3890 that have extents being checked for checksum errors, enter:

```
symchksum list -sid 3890
```

To show all the extents of device 0A1 on array 3890 that are being checked for checksum errors, enter:

```
symchksum show dev 0A1 -sid 3890
```

To enable Checksum on the extents of all the devices that define the current database instance and then to phone home on error, enter:

```
symchksum enable -type Oracle -phone_home
```

To enable Checksum on the extents of all the devices that define the tablespace and then to log on error, enter:

```
symchksum enable -type Oracle -tbs SYSTEM
```

To verify that Oracle tablespace USER01 has Checksum enabled on all the devices that have defined it, enter:

```
symchksum verify -type Oracle -tbs USER01
```

To disable Checksum on the current database instance, enter:

```
symchksum disable -type Oracle
```

Note: Disable by device should only be used under special circumstances. For example, this option can be used to remove extents if a database or a tablespace has been dropped without first doing a normal disable. In this case, disable by device can be used to remove the extents.

CHAPTER 20
Managing Network IPsec (Enginuity 5876 or lower)

This chapter describes the IPsec standard and how to manage IPsec network policies, using the IPsec component of the SYMCLI.

- Overview 560
- IPsec and EMC arrays 561
- SYMCLI IPsec management component 561
- Viewing policy information 562
- Command file scripting for creating/modifying policies 565
- Committing a policy-scripted command file 571
- Deleting a policy-scripted command file 571

Overview

IMPORTANT

IPsec is not supported on arrays running HYPERMAX OS 5977 or higher.

This section discusses the IPsec standard and how it applies to VMAX or DMX array networks.

IPsec is a framework of open standards for security of network communication that provides data confidentiality, integrity, and authentication between participating peers at the IP layer of an Ethernet network.

IPsec allows two peers (computers) to trust each other's identity and share a private key for each communication session. Based on the session key, the computer peers establish an encrypted communications channel and can verify that data they receive is what was originally sent. These countermeasures mitigate many security attacks, including spoofing, tampering with data en route, connection hijacking, eavesdropping, and replay of transmission. IPsec is the same security technology used in many corporate VPN solutions.

IPsec offers three main services to users:

- Authentication — Verifies that a packet was originated by the sender
- Integrity — Ensures that packets are not modified in transit
- Encryption — Hides the packet contents from prying eyes

Not all of these services are necessary or even desirable at the same time, in all situations. Users may selectively employ the ones that fit their current needs.

IPsec policies

An IPsec implementation internally manages database records called policies. They are processed in order, sorting on an attached index number called the policy's priority. In general, policies describe how to select applicable network traffic, and how to apply IPsec protection to it. They contain proposal and transform requirements for the IP tunnel.

More specifically, a policy is a combination of parameters that describe how network traffic will be encrypted. A policy is a single record in the policy database, and is indexed/sorted by Priority. When network traffic appears that might be affected by encryption, the IPsec implementation scans forward, in priority order, through the policy database looking for a policy that matches the traffic. Each policy has proposals and transforms that are also tested, in order. Once a match has been found, searching is terminated, and an entry is created in the Security Associations database that reflects the policy found. Also, each policy has a unique priority.

IPsec and EMC arrays

The VMAX or DMX array running Enginuity 5773 and higher, in companion with specific crypto processors, provides IPsec traffic services on each of its Gigabit Ethernet (Advanced Multi-Protocol channel) directors. Here, IP network traffic to or from the array can be protected by IPsec applied to it that supports two significant network connection protocols:

- SRDF — array to array endpoints, securing SRDF links
- iSCSI — iSCSI host to array endpoints or NAS to array endpoints

Traffic selection within IPsec is extremely fine-grained, as it is possible to apply IPsec only to iSCSI traffic with a connection to a particular host, or series of hosts, leaving other iSCSI traffic and SRDF untouched. Different destinations can have different types of protection.

Note: With such vast flexibility and versatility, configuration of IPsec is considered somewhat complex. It is recommended that IPsec management be performed with or by network engineers that understand encryption disciplines and the standard RFC 2401 for IPsec, including details of your specific network and environment. Incorrect intervention in this area, could expose critical information to third parties; especially if the two endpoints are connected with a publicly-visible network.

SYMCLI IPsec management component

The SYMCLI IPsec component extends the basic SYMCLI command set to include the Solutions Enabler `symipsec` command. This command allows you to perform all the necessary control operations that manage various types of IPsec security association policy changes in your VMAX environment. This IPsec standards compliant command provides policy management to/from the IPsec processor chip on the VMAX Gig-E port. It also offers retrieval of policies and associated statistics.

If you employ the Access Control feature with `symacl`, access types BASE and CFGSYM are required.

Information retrieved from the IPsec subsystem is not stored in the SYMAPI database. The IPsec policies are stored in global memory and backed up to the VMAX vault, therefore policy database backups and restores are not provided.

The `symipsec` command allows you to display and set values of control parameters for VMAX IPsec encryption and authentication support as follows:

- Lists various active IPsec policies.
- Shows detail about an active policy.
- Lists the various statistics and errors in the Enginuity processing of policies.
- Previews and commits the scripted command file that defines the specific network security policy requirements.
- Uses command file entries to define the IPsec policy requirements, including details such as proposals, and transforms.

Viewing policy information

With the `symipsec` command, you can return a list of IPsec policies or show detail about a policy being processed on a director and port.

Note: When specifying port numbers in the `symipsec` command, only port 0 is a valid parameter (and is the default). Specifically targeting Port 1 on a director is not supported.

Listing policies

To list information about policies on a specified VMAX array and director, use the following form:

```
symipsec -sid SymmID list -dir #|ALL -port 0
      -priority Level#
      -all
```

Where:

Level# — the policy level index number. The lower the number, the higher the priority level (0-110).

For example, to list the policy priority numbers on director 16D within a specific VMAX array, enter:

```
symipsec -sid 343 -dir 16D list port 0 -all
```

The output will list all current policies on the specified director:

```
Symmetrix ID                : 000190300343
Director number             : 16D
Port number                 : 0
Policy priorities present   :
    0         10        20        30        80        90
    100        110
```

Showing policy detail

To show detail information about policies on a specified VMAX array, use the following form:

```
symipsec -sid SymmID show -dir #|ALL -port #|ALL
      -priority Level#
      -all
```

For example, to show the policy details on director 1A for a specific policy (20), enter:

```
symipsec -sid 123456789012 -dir 1A -port 0 show -priority 20
```

For a second example, to show the policy details on director `16D` about all current policies, enter:

```
symipsec -sid 343 -dir 16D show -all
```

The output will show detail information about all current policies on the specified director, for example:

```
Symmetrix ID                    : 000190300343
Director ID                     : 16D
Port number                     : 0
```

```
Policy priority                 : 0
Action                          : Secure
Local Endpoint:
  Endpoint type                 : Single
  IP Version                    : v4
  IP Address                    : 10.0.0.40
  IP Port Number                : ANY
  IP Protocol Number            : ANY
  Flags                         : N/A
Remote Endpoint:
  Endpoint type                 : Single
  IP Version                    : v4
  IP Address                    : 10.0.0.51
  IP Port Number                : ANY
  IP Protocol Number            : ANY
  Flags                         : N/A
Source Port Selectivity         : Packet
Protocol Selectivity            : Policy
Destination Port Selectivity    : Policy
Proposal Set:
  Proposal Set Type             : IKE
  IN SPI                        : 0
  OUT SPI                       : 0
  IN CPI                        : 0
  OUT CPI                       : 0
  IN Nonce                      : 0
  OUT Nonce                     : 0
  IKE Mode                      : Main
  PFS                           : Off
  Number of Proposals           : 1
    Proposal #                  : 1
    Proposal type :             : IKE
    Number of Transforms        : 1
      Transform #               : 1
      Transform type            : IKE
      Authorization Algorithm   : MD5
      Encryption Algorithm      : 3DES
      Lifetime (bytes)          : 0
      Lifetime (seconds)        : 86400
      ESP Mode                  : Transport
      DH Group                  : DH Group 1
      Authentication Method     : Preshared Key
      Compression Algorithm     : N/A
Symmetrix ID                    : 000190300343
Director ID                     : 16D
Port number                     : 0
Policy priority                 : 10
Action                          : Secure
Local Endpoint:
.
.
.
```

The output pattern repeats for every policy found on the specified director.

Listing policy statistics

To list statistical information about the IPsec processing of a policy or policies, use the following form. You can optionally target a remote endpoint address:

```
symipsec -sid SymmID list -dir #|ALL -port 0
   -stats -type StatsType [-priority Level#][-remote_addr IPendPt]
```

Where:

StatsType — possible values are:

- ike_errors
- ipsec_details

IPendPt — the IP endpoint address

Level# — the policy level index number. The lower the number, the higher the priority level (0-110).

For example, to display IPsec SA statistical details for a specified director 2C, enter:

```
symipsec -sid 067 -dir 2c list -stats -type ipsec_details
```

The output will list statistical detail information about all current policies on the specified director:

```
Statistics from director 02C, port 0
                Local endpoint
  Endpoint type                : Single
  IP Version                   : v6
  IP Address                   : fd00:abcd:0:0:0:0:1:50
  IP Port Number               : 1748
  IP Protocol Number           : ANY
  Flags                        : N/A
                Remote endpoint
  Endpoint type                : Single
  IP Version                   : v6
  IP Address                   : fd00:abcd:0:0:0:0:1:41
  IP Port Number               : 48235
  IP Protocol Number           : ANY
  Flags                        : N/A

  SA Type                      : ESP
  ESP Mode                     : Transport
  Encryption Algorithm         : AES 128
  Authentication Algorithm     : XCBC
  Compression Algorithm        : N/A
  Total Rekeys                 : 0

                Inbound SA values
  SPI                          : 4025390203
  Byte Limit                   : 1099511627776
  Seconds Limit                : 5400
  Packet Count                 : 0
  Compressed Packet Count      : 0
  User Byte Count              : 0
  Compressed User Byte Count   : 0
  Authentication Errors        : 0
  Replay Errors                : 0
  Padding Errors               : 0

                Outbound SA values
  SPI                          : 4066949324
  Byte Limit                   : 1099511627776
```

```
  Seconds Limit               : 5400
  Packet Count                : 0
  Compressed Packet Count     : 0
  User Byte Count             : 0
  Compressed User Byte Count  : 0

Statistics from director 02C, port 0
                Local endpoint
  Endpoint type               : Single
```

The output pattern repeats for every policy found on the specified director.

To display IPsec IKE errors in the security processor on director 2C, enter:

```
symipsec -sid 067 -dir 2c list -stats -type ike_errors
```

The output will list statistical IKE or IPsec errors related to any of the current policies on the specified director, for example:

```
Statistics from director 34, port 0
  Endpoint type                 : Single
  IP Version                    : v6
  IP Address                    : fd00:abcd:0:0:0:0:1:50
  IP Port Number                : ANY
  IP Protocol Number            : ANY
  Flags                         : N/A
                Phase 1 (IKE) errors
  Init Failures :             : 0
  Response Failures :         : 0
  Invalid Cookies :           : 0
  No Peer Responses :         : 0
  Invalid Peer Responses :    : 0
  Logical Failures :          : 0
                Phase 2 (IPsec) errors
  Init Failures :             : 0
  Response Failures :         : 0
  Invalid Cookies :           : 0
  No Peer Responses :         : 0
  Invalid Peer Responses :    : 0
  Logical Failures :          : 0
```

Command file scripting for creating/modifying policies

The command and control of IPsec policy declarations are done through scripting and managing a command file. You can add, modify, or delete policies using various entries in a specified command file.

Command file syntax adding/modifying policies

The following is the command file syntax to add a network policy or modify an existing network policy. Leading pound signs (#) on a command line entry comments out that notation from the command parser:

Note: You can only define one proposal and one transform per policy declaration.

```
policy add|modify
   -priority Level#
   -action secure|discard|bypass
   [-assoc_ike_policy Level#]
   #only if proposal_type is ipsec
```

```
   -local_addr IPaddr
      [-ipproto IPprotocol#|all][-ipport IPport#|all][-mask IPaddr]
   -remote_addr IPaddr
      [-ipproto IPprotocol#|all][-ipport IPport#|all][-mask IPaddr]
   [-remote_tunnel_addr IPaddr
   #only if esp_mode is tunnel
   [-selectivity {destip|destport|srcport} SPECIFICITY]
   #for policies with wildcarded ipaddr/port/proto only

   -proposal_set
   -proposal_set_type auto|ike|manual
   [-ike_mode main|aggressive]
   #if proposal_set_type is set to ike mode
   [-pfs on|off]
   #if proposal_set_type is set to ike mode
   [-key_format hex |ascii]
   [-presharedkey Keystring]
   #if proposal_set_type is set to ike mode
   [-inenc_key Keystring -outenc_key Keystring]
   #if transform type is ESP or IKE and proposal is manual mode
   [-inauth_key Keystring -outauth_key Keystring]
   #if proposal is set to manual mode and algorithm is not null
   [-in_spi SPI# -out_spi SPI#]
   #if proposal is set to manual mode and algorithm is AES mode
   [-in_nonce NONCE -out_nonce NONCE]
   #if proposal is set to manual mode

   -proposal
   -proposal_type ike|ipsec

   -transform
   -transform_type ike|esp
   [-auth_alg null|sha1|md5|xcbc]
   [-enc_alg null|des|3des|aes_128|aes_256|aes_cm_128|aes_cm_256]
   #if transform type is esp or ike
   [-dhgroup 1|2|3|4]
   #if transform type is ike
   [-esp_mode tunnel|transport]
   #if transform type is esp
   [-lifetime [LifeParam1][,][LifeParam2]
   [-auth_method preshared_key|dsa|rsa]
   #if transform type is ike. DSA and RSA not supported.
;
```

Scripting the policy syntax

The beginning section of the policy syntax sets the overall policy requirements. For a policy add or modify task, consider and set the following options:

```
   -priority Level#
   -action secure|discard|bypass
   [-assoc_ike_policy Level#]
   -local_addr IPaddr
      [-ipproto IPprotocol#|all][-ipport IPport#|all][-mask IPaddr]
   -remote_addr IPaddr
      [-ipproto IPprotocol#|all][-ipport IPport#|all][-mask IPaddr]
   [-remote_tunnel_addr IPaddr
   [-selectivity {destip|destport|srcport} SPECIFICITY]
```

Where:

`-priority Level#`

Specifies the index number of the policy to be added or modified. Possible priorities range from 0 to 110. When packets arrive, policies with lower numbered priorities are examined first. Also, any associated IKE policies (`-assoc_ike_policy`) must have a lower priority index number (higher priority) than this object IPsec policy.

`-action secure|discard|bypass`

Sets the policy action to `secure`, requests that IPsec processing will be fully applied to packets that match this policy, requiring packet transformation.

Sets the policy action to `discard`, requests that the matching packets will be dropped completely. For use on unauthorized break-ins.

Sets the policy action to `bypass`, requests that the matching packets will be completely unmodified, and pass straight through the IPsec processor without encryption.

`-assoc_ike_policy Level#`

Specifies the index number of the associated IKE policy to the object IPsec policy. The `-proposal_type` option must be set to `ipsec`. When packets arrive, policies with lower numbered priorities are examined first. Associated IKE policies must have a lower priority index number (higher priority) than the corresponding IPsec policy.

```
-local_addr IPaddr
      [-ipproto IPprotocol#|all][-ipport IPport#|all][-mask IPaddr]
    -remote_addr IPaddr
      [-ipproto IPprotocol#|all][-ipport IPport#|all][-mask IPaddr]
```

These options primarily define the local and remote endpoint IP addresses. Optionally, you can target an IP protocol number (*`IPprotocol#`*) or all IP protocols and/or an IP port number (*`IPport#`*) or all IP port numbers. For example, IP protocol 6 should be used for TCP and 1 for IPv4-ICMP. And for example, IP port 1748 should be used for SRDF and port 3260 for iSCSI host connections. Note the SYMCLI can detect and handle both IPv4 and IPv6 address notations (dots and/or colons) in the command entry.

`-remote_tunnel_addr IPaddr`

Specifies the IP address of the remote tunnel object. The `-esp_mode` option must be set to `tunnel`.

`-selectivity {destip|destport|srcport} SPECIFICITY`

You can choose to use selectivity lists confined to destination or source points and can include *`SPECIFICITY`* types/protocols for wildcarded proposals only. When an endpoint field's properties have been wildcarded, it determines whether new connections will share an existing security association (selectivity `POLICY`), or if new connections will cause a new security association to be created (selectivity `PACKET`). Selecting `PACKET` results in a more secure configuration, since encryption keys won't be shared between connections, but consumes more resources. Selecting `POLICY` conserves security associations, when this is desired. Properties that may be wildcarded include IP address, IP port number, and IP protocol number.

Scripting proposal set syntax

The proposal set section of the policy syntax sets the proposal settings requirements. For a policy add or modify, consider and set the following options:

```
-proposal_set
-proposal_set_type auto|manual|ike
[-ike_mode main|aggressive]
[-pfs on|off]
[-key_format hex |ascii]
[-presharedkey Keystring]
[-inenc_key Keystring -outenc_key Keystring]
[-inauth_key Keystring -outauth_key Keystring]
[-in_spi SPI# -out_spi SPI#]
[-in_cpi CPI# -out_CPI CPI#]
[-in_nonce NONCE -out_nonce NONCE]
```

Where:

`-proposal_set`
`-proposal_set_type auto|manual|ike`

These options start the proposal set declaration. The `proposal_set_type` can be set to `auto`, `manual`, or `ike` for key management.

Automatic key management or SA management protocol is required to support anti-replay features of AH and ESP modes, and to accommodate on-demand creation of SAs. Multiple keys can be created as a result of this operation.

Manual key management is the simplest form of management where you must manually configure each system with keying data and security association management data relevant to secure communication with other systems. Manual techniques are practical in small, static environments, but it does not scale well.

The Internet Key Exchange (IKE) protocol is a key management protocol standard which is used in conjunction with the IPsec standard. IKE is a hybrid protocol which implements the Oakley key exchange and Skeme key exchange inside the SA and key management protocol (ISAKMP) framework. IKE eliminates the need to manually specify all the IPsec parameters in the crypto maps at both peer endpoints. It allows lifetime limits, changes to occur in keys during IPsec sessions, and allows anti-replay services and dynamic authentication of peers.

IKE options

`-ike_mode main|aggressive`
`-pfs on|off`
`-presharedkey Keystring`

These options require `proposal_set_type` to be set to `ike`.

IKE phase 1 negotiations are used to establish IKE security associations (SAs). These SAs protect the IKE 2 negotiations. IKE uses one of two modes for phase 1 negotiations: `main` or `aggressive`. Main mode is more intense and secure, but time consuming. Aggressive mode provides faster negotiations, but exposes identities of the peers to eavesdropping.

Perfect Forward Secrecy (PFS) mode for IKE policies can be turned on or off. (Typically, this should be left on, unless you have a special environment.

The preshared key option (`-presharedkey`) specifies the preshared key as the authentication method and defines the preshared key string. It should be the same secret string shared between security points.

Manual options

```
-inenc_key Keystring -outenc_key Keystring
-inauth_key Keystring -outauth_key Keystring
-in_spi SPI# -out_spi SPI#
-in_cpi CPI# -out_CPI CPI#
-in_nonce NONCE -out_nonce NONCE
```

These options require `proposal_set_type` to be set to `manual`.

For non-null `-enc_alg` only, the `-inenc_key` and `-outenc_key` options specify an encryption key string used for encrypting/decrypting inbound and outbound traffic. These must match the corresponding field on the remote endpoint.

For non-null `-auth_alg` only, the `-inauth_key` and `-outauth_key` options specify a hash key string used for authenticating inbound and outbound traffic. These must match the corresponding field on the remote endpoint.

The `-in_spi` and `-out_spi` options specify a Security Parameter Indexing (SPI) number for inbound and outbound traffic in security associations decode.

The `-in_nonce` and `-out_nonce` options specify a random nonce value for inbound and outbound traffic to counter replay attacks.

Proposal options

```
-proposal
-proposal_type ike|ipsec
```

These options start the proposal declaration. Currently, only one proposal per policy can be declared. Compression mode is not supported.

The `proposal_type` specifies the type of proposal to apply to the policy. Possible types supported are:

`ike` — IP Key Exchange

`ipsec` — IPsec

Transform options

```
-transform
-transform_type ike|ipsec
```

These options start the proposal declaration. Currently, only one transform per proposal and policy can be declared.

The `transform_type` specifies the type of transform to apply to the policy proposal. Possible types supported are:

`ike` — IP Key Exchange

`esp` — Encapsulation Security Payload

Currently, `ah` for authentication header is not supported.

`-auth_alg null|sha1|md5|xcbc`

Specifies the authentication algorithm for IKE or ESP policy and transform hash functions. Possible values are:

`null`

`sha1` — SHA-1 hash authentication algorithm

`md5` — MD5 hash authentication algorithm

`xcbc` — AES-XCBC-MAC constant key authentication algorithm

`-enc_alg null|des|3des|aes_128|aes_256|aes_cm_128|aes_cm_256`

Specifies the encryption algorithm for IKE or ESP policy transforms. Possible values are:

`null`

`des` — DES encryption algorithm

`3des` — 3DES or Triple DES (a.k.a. TDES) encryption algorithm

`aes_128` — Advanced Encryption Standard (AES) encryption (128-bit)

`aes_256` — Advanced Encryption Standard (AES) encryption (256-bit)

`aes_cm_128` — Advanced Encryption Standard counter mode (AES-CM)

`aes_cm_256` — Advanced Encryption Standard counter mode (AES-CM)

`-dhgroup 1|2|3|4`

Specifies which Diffe-Hellman (dh) group to use for the symmetrical key generation for IKE transforms only. Groups 1 through 4 are supported.

`-esp_mode tunnel|transport`

Specifies the ESP transform mode. Possible values are:

`tunnel` — Entire IP packet is encrypted/authenticated (full secure)

`transport` — Only the payload (just data) of the IP packet is authenticated

`-lifetime [`*`LifeParam1`*`][,][`*`LifeParam2`*

Specifies the life of the security policy with time and/or data size parameters for IKE and ESP only. Just one, or both values in any order, can be specified:

- A time value in minutes or hours (such as, 90 min or 5 hr)
- A data size value in megabytes or gigabytes (such as, 50 MB or 3 GB)

If both are specified, apply a comma between parameters with not intervening space. The first limit reached, will end the policy life.

`-auth_method preshared_key`

Specifies the authentication method for IKE transforms. Possible values are:

`preshared_key`

DSA and RSA methods are not supported.

Committing a policy-scripted command file

Once you have completed scripting the policy in the command file. You need to commit the file for execution. However, due to extensive notation work in any policy command file, it is recommended to preview the file for command syntax errors.

Preview the file

To preview the IPsec policy syntax of a command file, use the following command form:

```
symipsec -sid SymmID -dir # -port 0 -file /tmp/commandfile    preview
```

Note: When specifying port numbers in the `symipsec` command, only port 0 is a valid parameter (and is the default). Specifically targeting Port 1 on a director is not supported.

The following example, targets director 1A to preview an IPsec policy addition to the policy database, using file `/tmp/ap`:

```
symipsec -sid 0039 -dir 1A -port 0 -file /tmp/ap preview
```

Commit the file

To commit the IPsec policy syntax of a command file, use the following command form:

```
symipsec -sid SymmID -dir # -port 0 -file /tmp/commandfile commit
```

The following example, targets director 1A to commit an IPsec iSCSI policy addition to the policy database, using file `/tmp/ap`:

```
symipsec -sid 0039 -dir 1A -port 0 -file /tmp/ap commit
```

Where `/tmp/ap` contains:

```
policy add -priority 50 -assoc_ike_policy 40 -action secure
   -local_addr 172.23.195.20 -ipport 3260 -ipproto 6
   -remote_addr 50.60.70.80 -ipport 3260 -ipproto 6
   -selectivity destip packet -selectivity destport packet
   -proposal_set -proposal_set_type auto -proposal
   -proposal_type ipsec -transform -transform_type esp
   -enc_alg aes_cm_256  -lifetime 90m,5gb;
```

Deleting a policy-scripted command file

You can delete an existing IPsec policy.

To remove an IPsec policy from the policy database, enter:

```
symipsec -sid 0039 -dir 1A -port 0 -file /tmp/dp commit
```

Where `/tmp/dp` contains:

```
policy delete -priority 50;
```

CHAPTER 21
Events and Logs

This chapter describes how to configure, manage, and query VMAX events and logs.

- Overview 574
- Configuring log options 574
- Monitoring VMAX events using SYMCLI 576
- Common audit log 576
- Event daemon 579

Overview

SYMCLI and SYMAPI normally log significant events and actions to a daily log file. On UNIX, the log file has the following pathname:

```
/var/symapi/log/symapi-yyyymmdd.log
```

On Windows, the log file has the following pathname:

```
C:\Program Files\EMC\Symapi\log\symapi-yyyymmdd.log
```

where:

yyyy — year

mm — month

dd — day

The log displays the following items concerning each event:

- Time tag of the event occurrence
- Process ID (PID)
- Source of the event (application name)
- Related (internal) API function call
- Name of the specific operation or event
- Variable event field that describes the event or error in detail

Note: Log files accumulate over time and can consume needed disk space. Periodically, you may need to purge the log files to conserve space.

For a detailed list of possible VMAX events, refer to the *EMC Solutions Enabler SYMCLI Command Reference Guide*.

Configuring log options

Table 38 lists the options to customize how and where events are logged in your VMAX environment.

Table 38 Log file configuration options (page 1 of 2)

Logging option	Configuration description

Enable/disable logging	You can disable logging by setting the environment variable SYMCLI_NOLOGGING to 1. For example, to disable logging on UNIX (C shell) enter:
	`setenv SYMCLI_NOLOGGING 1`
	To turn logging back on, enter:
	`unsetenv SYMCLI_NOLOGGING`

Table 38 Log file configuration options (page 2 of 2)

Logging option	Configuration description

Log file name	You can change the name and path of the default log file.
	For example, to change the log file name on UNIX (C shell), use the following form:
	`setenv SYMCLI_LOG filename`
	
	To turn daily log files back on, enter:
	`unsetenv SYMCLI_LOG`

Allow undated SYMAPI log file names	You can allow the creation of undated SYMAPI log files by setting the environment variable SYMAPI_DATED_LOGFILE_NAME in the options file to `disable`.
	To allow undated log file names, set:
	`SYMAPI_DATED_LOGFILE_NAME=DISABLE`
	
	To reenable dates, set:
	`SYMAPI_DATED_LOGFILE_NAME=ENABLE`

Change date formats	You can change date formats in the log entries by setting the environment variable SYMAPI_LOGFILE_DATE_FORMAT in the options file to `FORMAT2`. This formats the date as *yyyy-mm-dd*.
	To change the date format, set:
	`SYMAPI_LOGFILE_DATE_FORMAT=FORMAT2`
	
	To change the date format back to the default (*mm/dd/yyyy*), set:
	`SYMAPI_LOGFILE_DATE_FORMAT=FORMAT1`

Log file configuration options	You can change format of optional fields within each log record by setting the environment variable SYMAPI_LOGFILE_FORMAT in the options file. Zero or more of the following optional fields can be included: • `pid` — include the process ID. • `tid` — include the thread ID. • `userid` — include the user ID (useful with User Authorization is enabled). • `activityid` — include the activity ID. The default value is `pid tid`. To see the log file format, create an entry in the options file with values separated by spaces. For example: `SYMAPI_LOGFILE_FORMAT = userid pid tid activityid`
Log file retention	Log file retention is an option set in the options file to specify the number of days to retain the log files: • Maximum value: 1825 (or 5 years) • Minimum value: 6 • Alternative value: 0 (setting it to zero maintains the log file forever. This is the default for everything except the service processor whose default is 30.) For example, to change the log file Retention to 60 days, enter: `SYMAPI_LOGFILE_RETENTION=60` **Note:** When the log file retention option is set, it overrides all existing log file Retention values including those set for the service processor during initialization. The default for the service processor is 30 days. Setting this options file value will override that default. In addition, if you have accumulated many years of log files prior to setting this option, when the option is set, it deletes those log files that do not meet the specified criteria.

Monitoring VMAX events using SYMCLI

The `symevent` command allows an administrator to monitor or track events within a VMAX array that may affect its operation. In some cases, reported events represent conditions that have already been repaired. This command allows the user to monitor the VMAX array for all reported events.

You can use `symevent` to either:

- `Monitor` — This action causes the command to run in the foreground, polling the VMAX array for new events every interval in seconds, either until the iteration count is satisfied or the program is stopped.
- `List` — This action examines the history of events, stored on the VMAX array, for those that meet the requested criteria. The `-start` and `-end` options allow the user to retrieve events that occurred between certain time bounds. In addition, you can sort the returned events based upon the reporting director using the `-DIR` option.

You can further refine your query to a VMAX ID and restrict the reported events to those of a minimum severity (warnings, errors, or fatal events).

Monitoring events

In its most basic form, `symevent` polls for and reports on all events, on all locally-connected VMAX systems, every 15 seconds, continuously:

```
symevent monitor
```

The following example polls for and displays events with a severity of warning or greater on array 0207 every 15 seconds for a 50-second period:

```
symevent monitor -sid 0207 -i 15 -c 50 -warn

Symmetrix ID: 000192600207

Detection time           Dir    Src  Category     Severity     Error Num
------------------------ ------ ---- ------------ ------------ ----------
Mon Dec 29 21:26:04 2008 DF-7A  Symm Communication Warning      0x001a
    The Symmetrix Service Processor could not complete a Call Home for service
```

Listing events

The following example illustrates how to retrieve a verbose list of the events which have occurred on the specified array between 12 p.m. and 3 p.m. today:

```
symevent list -sid 0207 -start 12:00 -end 15:00 -v
```

Common audit log

Data is written to a common audit file during VMAX control operations initiated by host applications. The common audit log correlates activity from all hosts into one file that is stored in the Symmetrix File System (SFS).

The `symaudit` command enables the examination and filtering of the common audit log file for a specified VMAX array. The audit log resides on the array and currently has a maximum size of 40 MB. Once the 40 MB limit is reached, the log begins to overwrite itself. You do not have to do anything to maintain this file, unless you want to capture records before the circular 40 MB space recycles.

Use the following actions to parse or monitor the audit log contents:

- `show` — Displays details about the audit log itself for a specific array, including: the total range of records, the date/time range, and the starting record number.
- `list` — Lists details about the requested records in either a brief or verbose format. The range of records to extract can be filtered by one or all of the following: record number, record count, date/time, functional area, control action, vendor ID, application ID, hostname, username, and device acted upon.
- `monitor` — Monitors the array for new audit log data in realtime.

Note: For details on valid parameter values on which to filter, refer to the `symaudit` command in the *EMC Solutions Enabler SYMCLI Command Reference Guide*.

Example output

The following is an example output of the `show` action for a specified array:

```
symaudit show -sid 0207

                A U D I T   L O G   D A T A

Symmetrix ID            : 000192600207

Starting date           : 12/17/2008 21:42:57
Ending date             : 01/02/2009 15:30:35

Starting record number  :       1
Ending record number    :    1737
Total record count      :    1737
```

The following is an example output of the `list` option for a specified array between a specified time period:

```
symaudit list -sid 0207 -start_date 01/02/2009:12:00:00 -end_date
   01/02/2009:12:15:00 -v

              A U D I T   L O G   D A T A

Symmetrix ID            : 000192600207

  Record Number         :     1663
    Records in Seq      :        2
    Offset in Seq       :        1
    Time                : 01/02/09 12:11:50
    Vendor ID           : EMC Corp
    Application ID      : SYMCONFIGURE
    Application Version : 7.0.0.302
    API Library         : SDK
    API Version         : T7.0.302.2 (Edit Level: 907)
    Host Name           : api1051.lss.
    OS Name             : LINUX
    OS Revision         : 2.6.9-11.E
    Client Host         :
```

```
    Process ID          : 00027021
    Task ID             : 00000002
    Function Class      : CfgChg
    Action Code         : Release
    Text                : STARTING a Device Reservation 'RELEASE'. Owner=m; Rese
rveID=1; Comment="t";
    Username            : H:api1051\root
    Activity ID         : SE69f82805ea

  Record Number         :     1664
    Records in Seq      :        2
    Offset in Seq       :        2
    Time                : 01/02/09 12:11:50
    Vendor ID           : EMC Corp
    Application ID      : SYMCONFIGURE
    Application Version : 7.0.0.302
    API Library         : SDK
    API Version         : T7.0.302.2 (Edit Level: 907)
    Host Name           : api1051.lss.
    OS Name             : LINUX
    OS Revision         : 2.6.9-11.E
    Client Host         :
    Process ID          : 00027021
    Task ID             : 00000002
    Function Class      : CfgChg
    Action Code         : Release
    Text                : Devices: [ 0020 ]
    Username            : H:api1051\root
    Activity ID         : SE69f82805ea

  Record Number         :     1665
    Records in Seq      :        1
    Offset in Seq       :        1
    Time                : 01/02/09 12:11:51
    Vendor ID           : EMC Corp
    Application ID      : SYMCONFIGURE
    Application Version : 7.0.0.302
    API Library         : SDK
    API Version         : T7.0.302.2 (Edit Level: 907)
    Host Name           : api1051.lss.
    OS Name             : LINUX
    OS Revision         : 2.6.9-11.E
    Client Host         :
    Process ID          : 00027021
    Task ID             : 00000002
    Function Class      : CfgChg
    Action Code         : Release
    Text                : Device Reservation 'RELEASE' SUCCEEDED. ReserveID=1;
    Username            : H:api1051\root
    Activity ID         : SE69f82805ea

<. . .>
```

- Record Number — The current record number.
- Records in Seq — Total number of records requested.
- Offset in Seq — Offset number from the first record requested.
- Time — Date and time the record was entered.
- Vendor ID — ID of the vendor whose application logged the record.
- Application ID — ID of the application that logged the record.
- Application Version — Application version number.

- API Library — Name of the SYMAPI library the application ran against.
- API Version — Version of the SYMAPI.
- Host Name — Name of the host that logged the record.
- OS Name — Operating system on which the host is running.
- OS Revision — Operating system revision number.
- Client Host — Any SYMCLI client communicating with the SYMAPI server.
- Process ID — ID of the process that logged the record.
- Task ID — ID of the task.
- Function Class — Class name of the SYMAPI functional area.
- Action Code — Name of the SYMAPI control action associated with an audit log entry.
- Text — Details of the given entry.
- Username — Identifies the user that generated the log entry.
- Activity ID — A randomly generated ID that uniquely identifies this action.

Event daemon

Solutions Enabler also provides an asynchronous event daemon that can be used to monitor and manage VMAX events. The event daemon (`storevntd`) provides the necessary services required to monitor the status of VMAX storage environments from third-party enterprise management frameworks. The following targets are supported:

- SNMP
- File on disk
- System logger on the host
- Unix syslog service
- Windows event log
- The syslog listener across the network (bypasses the syslog service (calls) on the local host and directly sends events/traps to this remote listener).

To learn more about enabling and configuring the event daemon, refer to the *EMC Solutions Enabler Installation Guide*.

CHAPTER 22
XML Structured Output

This chapter describes the XML output option of the SYMCLI.

- Overview 582
- XSLT: XML data transformations 582
- Element-based XML 582
- Using XML mode 583

Overview

The XML (Extensible Markup Language) output option provides a mechanism to facilitate the automated processing of SYMCLI output data. XML is a deterministic parsing tool that eases parsing of output data, providing a functional advantage over screen-scraping tools like awk or Perl. The XML industry standard is based on the experience of SGML and is endorsed by the World Wide Web Consortium. Detailed information on XML may be found at:

http://w3.org/XML/

XML has the look and feel of HTML, as it employs the same tag-based syntax. However, XML uses tags to delimit data—as opposed to defining the data as with HTML—allowing the document author to specify the tags most applicable to the given application. For the SYMCLI, tags represent the physical and logical structures within the VMAX array and its environments.

When XML mode is utilized, the data returned is identical to that of the *standard* output, but “marked-up” with tags. These tags enable individual pieces of data to be readily called upon by name. In addition, they provide a definitive way to express the relationship between different objects, an advantage over the standard CLI display output.

XSLT: XML data transformations

Many tools are available to query, filter, retrieve, and format specific information stored in complex XML files. Among these, eXtensible Stylesheet Language Transforms (XSLT) is a particularly useful and widely available technology. While using XML will result in less ambiguous, more robust scripts, XSLT will make the information presented in XML accessible to the more familiar plain text-based scripting techniques. To introduce you to XSLT, a directory containing several examples of the types of queries that can be performed on XML data using XSLT is provided. The examples are designed only to provide a brief introduction to the power and usefulness of XSLT, and can help ease the transition to XML.

Element-based XML

Solutions Enabler provides element-based XML that describes data in a hierarchical manner by using the notion of parent and children. An element can have several different content types. It can have element content (child element), a mixed content containing both text and child element, a simple content containing text only, or an empty content carrying no information. An element can also have attributes. These additional content types would allow users to modify the data structures in a fairly flexible manner. On the other hand, an attribute is used to provide additional information about an element. An attribute is in general used to store the metadata describing the data that stored in XML. Although data can be stored in attributes, it is best practice to store data in child elements.

Using XML mode

To use XML mode with SYMCLI, an environment variable or a command line option can be used. Use the environment variable to globally set your command output to XML or globally return it to standard mode. Use the following syntax to set the environment variable:

```
SYMCLI_OUTPUT_MODE = xml_element|standard
```

- **xml_element** — Returns the output of all commands in element-based XML tags.
- **standard** — Returns the output of all commands to the default output without the XML tags.

Note: When the environment variable output mode is set to `xml`, `xml_element`, commands that do not support XML output generate a runtime error message. You can override this behavior by setting the command line `-output` option to a value of `standard`. This allows you to successfully execute the given command in standard mode.

The command line method provides a means of outputting any single command as either XML or standard output. Using the command line method overrides the current environment variable setting:

```
<SymcliCommand> -output <xml|xml_element|standard>
```

The `-output` flag is not found in `-help` or man pages because of its wide scope and usage.

Example output

Notice that a new element tag `<Symm_Info>` was added to the XML data shown below to store general VMAX data.

`symcfg list -out xml` OR **`symcfg list -out xml_element`**

```
<?xml version="1.0" standalone="yes" ?>

<SymCLI_ML>
  <Symmetrix>
    <Symm_Info>
      <symid>000190102055</symid>
      <attachment>Local</attachment>
      <model>DMX3-24</model>
      <microcode_version>5772</microcode_version>
      <cache_megabytes>32768</cache_megabytes>
      <devices>683</devices>
      <physical_devices>94</physical_devices>
    </Symm_Info>
  </Symmetrix>
  <Symmetrix>
    <Symm_Info>
      <symid>000190300215</symid>
      <attachment>Local</attachment>
      <model>DMX3-6</model>
      <microcode_version>5772</microcode_version>
      <cache_megabytes>32768</cache_megabytes>
      <devices>239</devices>
      <physical_devices>2</physical_devices>
    </Symm_Info>
```

```
    </Symmetrix>
    <Symmetrix>
      <Symm_Info>
        <symid>000190300237</symid>
        <attachment>Remote</attachment>
        <model>DMX3-6</model>
        <microcode_version>5773</microcode_version>
        <cache_megabytes>16384</cache_megabytes>
        <devices>3756</devices>
        <physical_devices>0</physical_devices>
      </Symm_Info>
    </Symmetrix>
    <Symmetrix>
      <Symm_Info>
        <symid>000190300343</symid>
        <attachment>Remote</attachment>
        <model>DMX3-6</model>
        <microcode_version>5773</microcode_version>
        <cache_megabytes>32768</cache_megabytes>
        <devices>751</devices>
        <physical_devices>0</physical_devices>
      </Symm_Info>
    </Symmetrix>
  </SymCLI_ML>
```

To maintain consistent element names in all SYMCLI commands, some tag names are redefined. For example, in current SYMCLI command, different names exist to describe Symmetrix identification number, such as id, symmetrix, or symid. A new consistent tag name is defined across all SYMCLI commands in the element-based XML output.

PART 3
Migrating Data

Part 3 of this guide describes how to use SYMCLI Open Replicator to copy data between arrays and SYMCLI Federated Live Migration to migrate data from DMX arrays to VMAX arrays.

Chapter 23, "Open Replicator Operations" introduces Open Replicator and explains how to perform copy operations.

Chapter 24, "Operational Restrictions and State Reference" provides the rules for using Open Replicator with other replication operations and devices.

Chapter 25, "Federated Live Migration Operations (Enginuity 5876 or lower)" provides an operation overview of FLM, example CLI commands that support FLM, and example reporting commands.

CHAPTER 23
Open Replicator Operations

This chapter introduces the EMC Open Replicator SYMCLI command (`symrcopy`) and explains how to implement this command for copying device data between arrays across the storage network.

- Open Replicator overview 588
- Open Replicator command summary 596
- Open Replicator operations 597
- Using Open Replicator to perform a migration 600
- Open Replicator examples 622

Open Replicator overview

The EMC Open Replicator command (`symrcopy`) provides a method for copying device data to or from various types of arrays within a storage area network (SAN) infrastructure. For example, Open Replicator provides a tool that can be used to migrate data from older arrays, and some third-party storage arrays to a VMAX Family array. The `symrcopy` command is part of the SYMCLI command library.

The Open Replicator command can also be used to migrate data from a DMX or VMAX Family array to other types of storage arrays within the SAN infrastructure. Copying data from a DMX or VMAX Family array to devices on other storage arrays allows for data to be copied fully or incrementally.

Open Replicator RecoverPoint sessions can be listed or terminated using the `symrcopy` command, however RecoverPoint sessions cannot be created or modified through this command.

⚠CAUTION

Data migrations are often complex operations and require careful planning and execution of predetermined procedures. Failure to identify and perform necessary steps or work within supported configurations can result in data unavailability or loss.

Note: For detailed interoperability information, please refer to the E-Lab Interoperability Navigator available at http://elabnavigator.EMC.com.

Practical uses

The `symrcopy` command can be used for:

- Migrating data between DMX, VMAX Family arrays, and third-party storage arrays within the SAN infrastructure without interfering with host applications and ongoing business operations.
- Backing up and archiving existing data within the SAN infrastructure as part of an information lifecycle management solution.
- Federated Live Migration operations. FLM allows device copying, from donor DMX arrays and VMAX 20K arrays to new VMAX 10K, 20K, and 40K arrays, without restarting the application hosts. Refer to Chapter 25, “Federated Live Migration Operations (Enginuity 5876 or lower)” for more information.

  Note: For arrays running HYPERMAX OS 5977 or higher, FLM is not supported.

- Listing and terminating RecoverPoint sessions.

Limitations for HYPERMAX OS 5977 or higher

The following limitations apply to arrays running HYPERMAX OS 5977 or higher:

Since ORS push sessions are not supported, the following operations, allowed only for push sessions, are not supported:

- Differential copying
- Precopy option

- Recreate command
- Restore command
- Remove command

- The maximum number of active sessions allowed is 512 when running HYPERMAX OS 5977 or higher.
- Only one remote target is supported.
- Creating ORS sessions on control devices that are involved in a TimeFinder session or a SRDF operation is not supported.

Functionality

Open Replicator copy (Rcopy) operations are controlled from a local host attached to the DMX or VMAX Family array. Data copying is accomplished as part of the storage system process and does not require host resources. The data can be copied online between DMX and VMAX arrays allowing host applications, such as a database or file server, to remain operational (function normally) during the copy process.

Rcopy concepts

The following Rcopy concepts and terminology are used throughout the Open Replicator Migration section of this product guide:

- The VMAX Family and DMX arrays and its devices are referred to as the *control* side of the copy operation. Older DMX or VMAX arrays, or third-party arrays on the SAN are referred to as the *remote* array/devices.
- The copy direction is always from the perspective of the control side. There are two types of copy operations, *push* and *pull*. A push operation copies data from the control device to the remote device(s). A pull operation copies data to the control device from the remote device(s).
- Copy operations are either *hot* (online) or *cold* (offline).
- Use the `-name` option to give the session a name. Use the `-session_name` option when specifying the session name for control operations.
- There can be only one control device per active session.

Open Replicator can be used to migrate data into a VMAX Family array from older DMX and VMAX arrays, or other third-party arrays. Figure 22 shows two Open Replicator copy sessions performing a *pull* operation, where data is copied through the SAN infrastructure from remote devices to the control array.

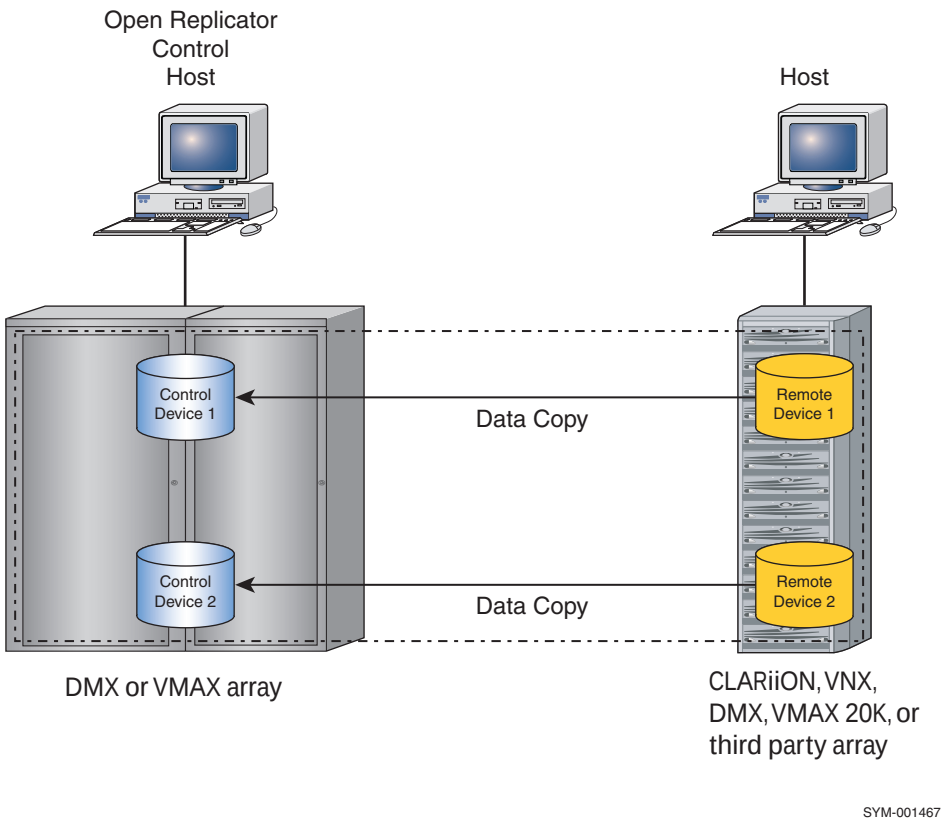


Figure 22 Control array device pull operation

Note: Since data is copied through the SAN infrastructure, Open Replicator may require updating the zoning configuration before copying data between arrays is allowed. For zoning requirements and suggestions, refer to "SAN setup requirements" on page 594.

Open Replicator can be used to copy data from a control array to older array. Figure 23 shows two Open Replicator copy sessions performing a *push* operation, where data is copied from the control array to remote devices within the SAN infrastructure.

For arrays running HYPERMAX OS 5977, push operations are not supported.

For

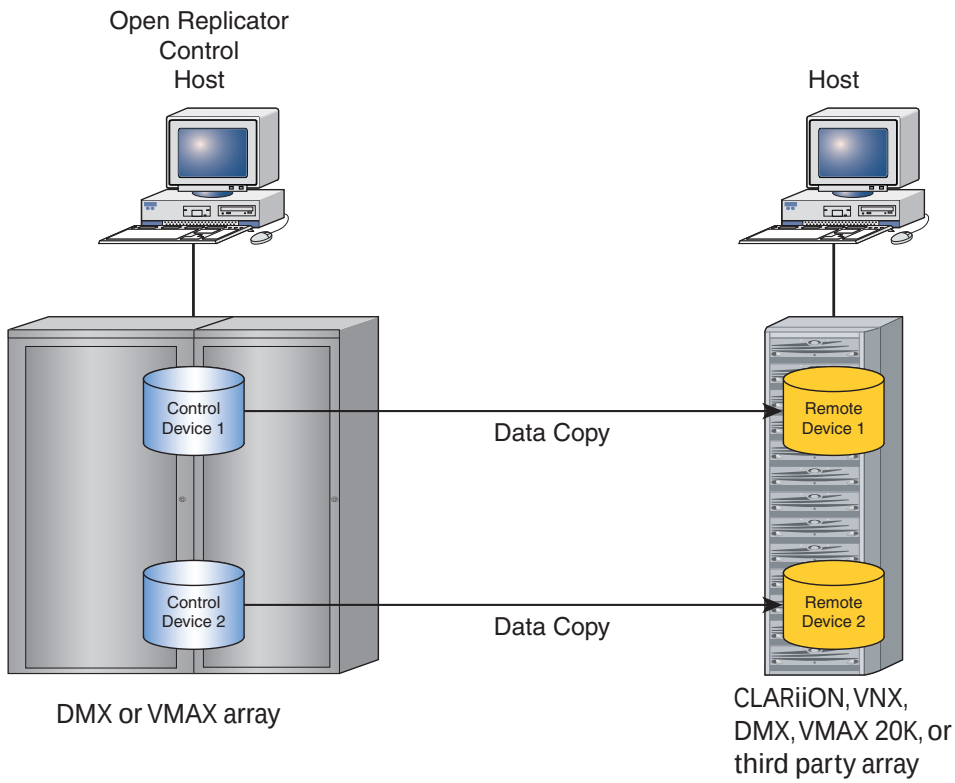


Figure 23 Control array device push operation

Operational rules and limitations

The following general rules apply to Open Replicator sessions:

- Remote devices do not have to be the same RAID type or meta-configuration.
- For a push operation, remote capacity must be equal to or larger than the control device extents and vice versa for a pull operation. The exception to this is when data has been pushed to a remote device that is larger than the control device, and the data needs to be pulled back to the control device. In this case, use the `-force_copy` option.
- On push copy operations, remote devices should not be accessed by their remote hosts until copying has completed.
- On pull operations, the remote devices should not be updated by array hosts for the duration of the copy process.
- For pull operations from devices with SCSI reservations, if the remote devices have a cluster running against them or the devices are AIX LVM devices, the cluster, AIX host, or other software that is creating the SCSI reservations must be shutdown before creating the Open Replicator session.

- Data corruption to devices is possible during a copy operation if another host on the SAN has write access to the remote device. If this is a concern, EMC recommends that the remote device be unmounted or marked as Not Ready to any other hosts on the SAN to guarantee that the device cannot change while copying is in process.
- Accumulated I/O errors between the control device and remote device will cause a session to fail if the copy operation is a hot push. The failed session may be activated again as long as no new data has been written to the control device since the session failed. The session will temporarily stall and restart on any other type of copy operation.
- Open Replicator fully supports copy operations for thin devices. For information on Virtual Provisioning™ and creating thin devices, refer to "Configuring Virtual Provisioning" on page 275.
- Open Replicator supports copy operations for a cold push from a virtual device (VDEV). For more information and instructions, refer to "Using VDEVs in Open Replicator" on page 620.

Limitations for Enginuity 5876 or lower

The following limitations apply to arrays running Enginuity 5876 or lower:

- Up to 15 incremental Open Replicator copy sessions can exist for a single array device.
- The maximum number of active sessions allowed is 1024 when running Enginuity version 5773 through 5876, with the default set to 1024.
- The maximum number of active sessions allowed is 512 when running Enginuity version 5772 or lower, with the default set to 512.

  Note: To change the session limit, use the SYMAPI_RCOPY_SESSION_LIMIT option.

- The maximum concurrent remote devices is 16 per session.
- Only one copy session per control device can be active at a time. The active session must be completed or terminated before another session can be created or started on a device.
- An Open Replicator copy session is between two arrays and cannot be created with control and remote devices on the same array. Copying within the same array must be done using SYMCLI control operations, such as TimeFinder/Snap and Clone.
- Devices that are accessible only through the SAN cannot be specified with array device names. Any World Wide Name (WWN) of a remote system that is more than two RDF links away must be obtained using platform-native tools.
- On an offline push operation to multiple devices that were created as part of one session, devices in the session must be updated together. A single device cannot be updated independently.
- Open Replicator can be used only with FBA devices using Fibre Channel integrated directors. The following device types, that could be mapped to the host, are not supported:
  - CKD
  - VCM

- WORM
- Metamembers

- Only one remote device is allowed for a hot push operation.
- Open Replicator copy sessions involving older arrays (prior to the DMX) require a connection through a switch to copy data.
- Third party or non-visible storage systems cannot be validated by the Solutions Enabler.

Limitations for copying

Copying is device-based; extent copying is not supported. Device configuration changes cannot be made during an Open Replicator session, as making device changes may lead to inconsistent data on the local device if pulling, or on the remote device if pushing data.

Open Replicator cannot detect changes to a remote device during, or between incremental copies. Before each session, make sure that there are no changes being made to the remote device.

Federated Tiered Storage support

There is support for Federated Tiered Storage devices with Open Replicator operations, however there are some limitations as listed below:

- With Federated Tiered Storage (FTS), an external LUN (eDisk) is attached through the SAN to an array, and is used as an external back-end disk for the array.
- FTS requires a new front end disk director (DX) and Enginuity 5876 or higher.
- Adding an eDisk through the SAN to the array provides the ability to migrate user data from the external storage into the array. There are two modes for presenting the eDisk to the array, *external provisioning* and *encapsulation*.

For more information on FTS refer to "Federated Tiered Storage" on page 301.

- FTS rules and restrictions with Open Replicator:
  - Externally provisioned devices are fully supported.
  - Encapsulated devices that are not geometry limited are fully supported.
  - Encapsulated devices that are geometry limited devices are not supported for `-pull` operations.

Device guidelines

Table 39 provides some guidelines for control and remote devices.

Table 39 Control and remote device guidelines

Action	Control device	Remote device
Creating the device file	DMX or VMAX Family arrays Always listed on left Format: symdev=arrayid:device Example: symdev=7098:E9	DMX or older VMAX array, or third-party array Always listed on right Format: symdev =array:device or wwn=WWN Example: wwn=60060480000000000314353594D303737 Refer to "Creating a device file" on page 601 for device format rules
Hot push (not supported on HYPERMAX OS 5977)	One device per session All directors must see remote device	One device per session Device not accessible to host Can use `-consistent`, `-nodifferential`, and `-precopy` **Note:** Differential copying is the default for copy sessions and does not need to be specified in the `symrcopy create` command.
Cold push (not supported on HYPERMAX OS 5977)	One device per session Device Not Ready to host At least one director must see remote device(s)	Can use `-nodifferential` **Note:** Differential copying is the default for copy sessions and does not need to be specified in the `symrcopy create` command.
Hot pull	One device per session Device online to the host All directors must see the remote device[a]	One device per active session Can use `-donor_update`, `-frontend_zero`
Cold pull	One device per session At least one director must see the remote device and the devices is not ready to host	One device per session Can use `-frontend_zero`

a. Applies only to devices running on arrays Enginuity 5876 or lower. This is not a requirement for HYPERMAX OS 5977 or higher.

SAN setup requirements

Since data is copied through the SAN infrastructure, Open Replicator may require a SAN configuration update before copying data between storage arrays is allowed. Because of the various types of cabling, zoning, and masking that can exist within a SAN configuration, the following requirements are provided as a generic reference for setting up a data migration with Open Replicator through a SAN:

- A Fibre Channel switch is required for Open Replicator. Direct connections (such as arbitrated loop) are not supported.
- SAN setup for arrays running Enginuity 5773 through 5876

- The SAN for the remote storage array must have connectivity to the control array SAN. Open Replicator requires that at least one port on the remote array, that allows access to the remote device, have access to the control device through at least one port for a cold copy and all ports for a hot copy on the control array.
- Zoning must be set up on the Fibre Channel switch to zone the control array fibre adapters (FAs) to the remote storage array front-end adapter(s).
- If the storage port for the source devices is running volume configuration management (VCM) software such as Volume Logix for Symmetrix or a similar software product, setup may require granting permission to access the target device(s) from the target storage port. For instance, on a VMAX array, the control FA(s) must be enabled to have access to the remote storage array device(s).
- Specific instructions for granting access to target devices vary depending on the selected target storage array and VCM software.

◆ Arrays running HYPERMAX OS 5977 will dynamically find directors and ports that can access an ORS session's remote devices, and it will use these directors and ports for the data transfer. HYPERMAX OS 5977 requires at least one zoned remote port for cold sessions, and it is recommended that there are at least two zoned remote ports for hot sessions. The recommendation for hot sessions is not enforced.

SYMCLI symsan support

The SYMCLI command `symsan` lists port and LUN WWNs as seen from a specific array director and port. This can be used to validate that the zoning between the port and target is correct. It does not require a created Open Replicator session. Use this command to display remote port WWNs, and LUN WWNs seen behind a remote port WWN.

The `symsan` command allows for:

◆ Listing all ports or LUNS in the SAN that can be seen by a specific DX director or all DX directors.

◆ Listing all ports or LUNS in the SAN that can be seen by a specific FA director or all FA directors.

◆ Listing all ports or LUNS in the SAN that can be seen by a specific FA/DX director or all FA/DX directors, using the `-dir` option.

Refer to "Getting SYMCLI help" on page 29 for information on how to access `symsan` manpage or command help.

Open Replicator command summary

Table 40 summarizes the SYMCLI commands used to manage an Open Replicator copy session.

Table 40 SYMCLI Open Replicator command summary

Command	Argument	Displays
symrcopy	activate	Starts the copying process on an existing Open Replicator copy session.
	create	Defines a new Open Replicator copy session.
	export	Creates the file (*FileName* given) with all of the session information for sessions matching the session name.
	list	Lists all Open Replicator copy sessions for a given control array.
	query	Queries for the status of Open Replicator copy sessions.
	recreate[a]	Creates an incremental copy session on an existing copy session. Only valid for push copy sessions created with differential copying.
	remove[a]	Removes remote devices from a differential push session in the copied state. Not supported for arrays running HYPERMAX OS 5977 or higher.
	rename	Changes the name of a session. Can be executed only on a differential session in the Copied state.
	restore[a]	Restores data from a remote device that was previously copied during a differential push operation back to the control device.
	set donor_update off	Turns off the donor update option for the session. When set donor_update off is used with the -consistent option, the consistency of data on the remote devices will be maintained.
	set frontend_zero off	Turns off the front-end zero detection option for thin control devices for the session.
	set mode	Sets the mode for copying to either CopyInProg, CopyOnAccess, CopyOnWrite, or Precopy.
	failback	Used with the -migrate option to stop a Federated Live Migration session.
	set pace[a]	Sets the session pace for the CopyInProg, RecreateInProg, and RestInProg states. Possible value are 0-9, with 9 being the slowest pace.
	terminate	Terminates a copy session and removes it from the control array.
	verify	Verifies that a copy session is in an existing state.
symqos -rcopy	set ceiling	Sets the maximum allowed bandwidth percentage for a given director, port, director, and port, or all directors and ports. Acceptable values are 0-100 or NONE to shut the ceiling function off. For arrays running HYPERMAX 5977 or higher, ceiling can be set to DISABLE. This will block Open Replicator I/O on the specified director/port.
	list ceiling	Lists the maximum bandwidth percentage for a given director, port, director/port pair, or all directors and ports.

a. Not supported for arrays running HYPERMAX OS 5977 or higher.

Open Replicator operations

Open Replicator copies data in sessions across the SAN infrastructure. A device file is used to specify the device pairs to be used in the copy session. These devices are referred to as the *control* and *remote* devices. The control device always resides on the locally-attached DMX or VMAX Family array, and is responsible for controlling data copying to or from its partner remote device. Devices listed in the device file are identified by either logical unit number (LUN), World Wide Name (WWN), or by a combination of the storage array ID and device name (use `symdev` for arrays). Refer to "Creating a device file" on page 601 for instructions on how to obtain device information and create the device file.

A copy session is first defined by using the `symrcopy create` command. A session name can be specified for later use in control operations. The push/pull options (`-push`|`-pull`) are used to define the direction of the copy operation for device pairs listed in the device file. If the copy direction option is set as a pull operation, data will be pulled in through the SAN to the control device(s) from the remote device(s). If the copy direction option is set as a push operation, data will be pushed across the SAN from the control device(s) to the remote device(s).

A copy session is also defined as either *hot* (online) or *cold* (offline) copying by using the `-hot` or `-cold` option parameters. Hot copying allows the control device to be read/write online to the host while the copy operation is in progress. With hot copying, all directors that have the local devices mapped are required to participate in the session. A hot copy session cannot be created unless all directors can discover the remote device.

Note: Arrays running HYPERMAX OS 5977 dynamically find directors and ports that can access an ORS session's remote devices, and use these directors and ports for data transfer.

With cold copying the control device is write disabled to the host while the copy operation is in progress. A cold copy session can be created as long as one or more directors discovers the remote device.

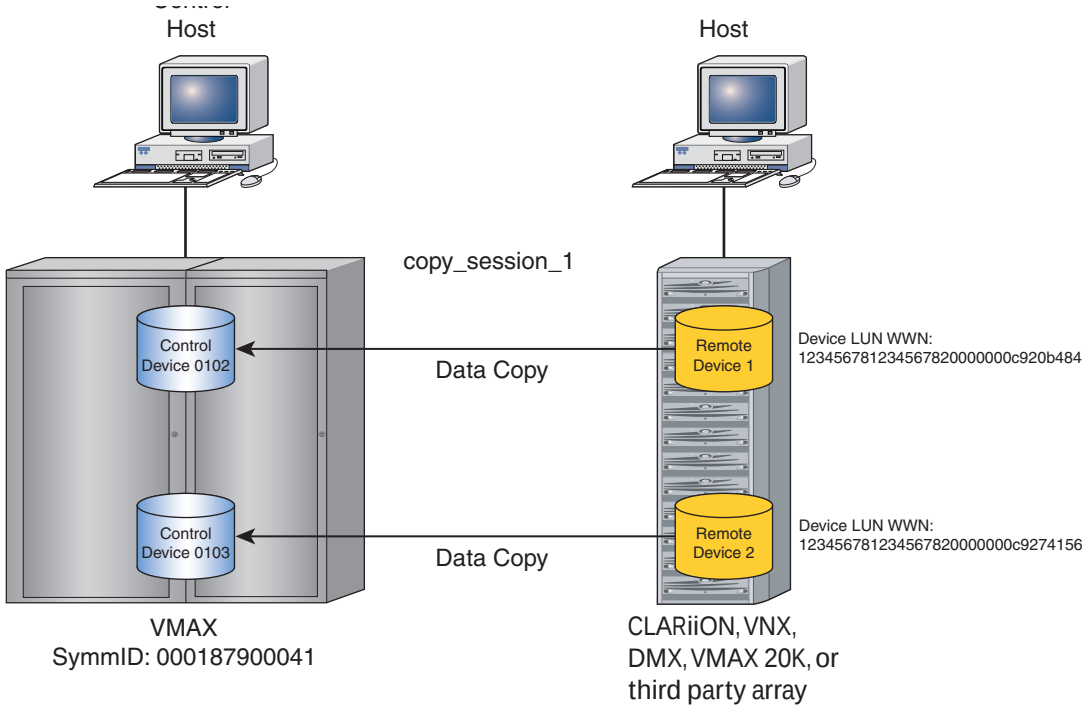
If a control device is pushing data to a remote device and that control device is currently online for host write I/O operations, a consistent point-in-time copy can be made across multiple control devices using the Enginuity Consistency Assist (ECA) feature (`-consistent`). This will temporarily prevent any host write I/Os while the Open Replicator copy session begins.

Hot pull copy session example

Figure 24 on page 598 shows Open Replicator copy sessions created and activated for a hot pull copy operation. The device file (`-file`) contains the pairing information for the control and remote devices. Each line in the device file is a copy session. Remote devices in the file are specified by "LUN WWN" and control devices are specified by "Symmetrix ID: device number" as follows:

```
symdev=000187900041:0102   wwn=1234567812345678200000000c920b484
symdev=000187900041:0103   wwn=1234567812345678200000000c9274156
```

Note: Refer to "Creating a device file" on page 601 for instructions on how to obtain device information and create the device file.




```
symrcopy create -name copy_session_1 -pull -hot -file pairs
symrcopy activate -file pairs
```

Figure 24 Control array hot pull using the **`symrcopy`** command

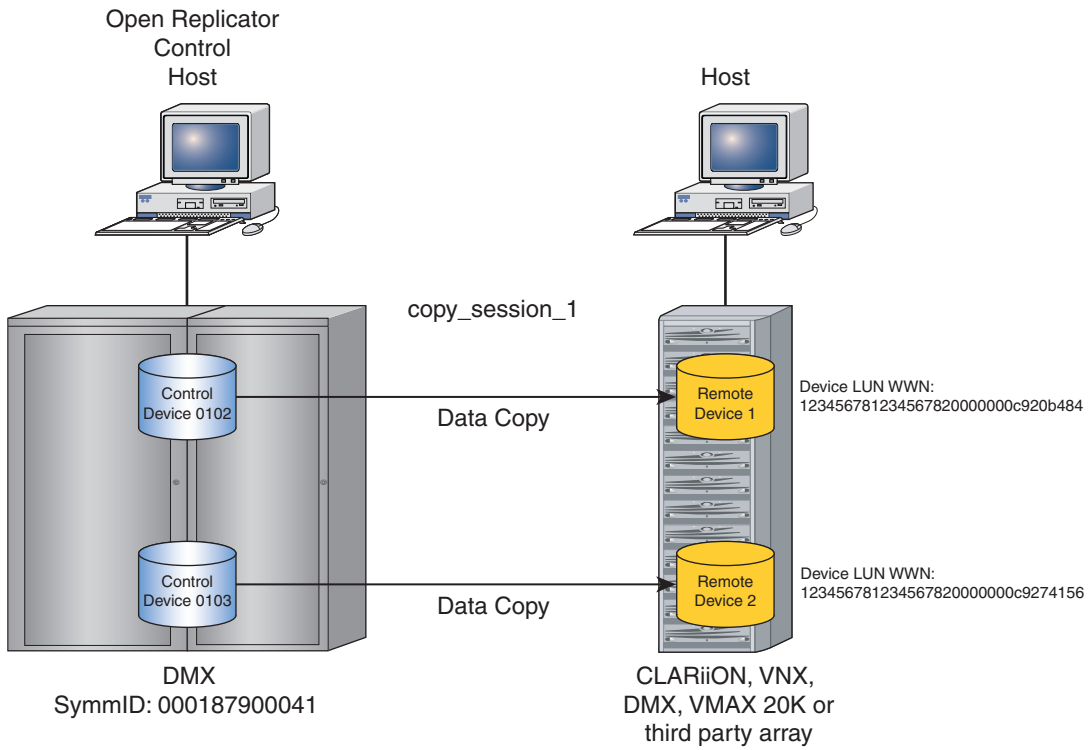
Cold push copy session example

Figure 25 on page 599 shows Open Replicator copy sessions created and activated for a cold push operation. The device file (`-file`) contains the pairing information for the control and remote devices. Each line in the device file is a copy session. Control devices are specified by "Symmetrix ID: device number" and remote devices in the file are specified by "LUN WWN" as follows:

```
symdev=000187900041:0102   wwn=1234567812345678200000000c920b484
symdev=000187900041:0103   wwn=1234567812345678200000000c9274156
```

Note: Refer to "Obtaining device information" on page 601 and "Creating the file" on page 602 for instructions on how to obtain device information and create the device file.

Note: For arrays running HYPERMAX OS 5977, push operations are not supported.




```
symrcopy create -name copy_session_1 -push -cold -file pairs
symrcopy activate -file pairs
```

SYM-001470

Figure 25 Control array device cold push using the **symrcopy** command

Using Open Replicator to perform a migration

SYMCLI Open Replicator control operations are performed from a local host attached to the DMX or VMAX Family array and are implemented in sessions across the SAN infrastructure. Open Replicator copy sessions are first created using a device file, which lists the device pairs (control and remote) for the operation.

Open Replicator uses the SYMCLI `symrcopy` command to perform copy sessions. The main control operations required to successfully complete a copy session are as follows:

- Create the session.
- Activate the session.
- Terminate the session.

For arrays running HYPERMAX OS 5977, the following optional settings and actions are available:

- Data protection and recovery options for hot and cold pulls.
- Enable or disable front-end zero detection for pull operations to thin control devices.
- Background copying mode of a session.
- Ceiling value for bandwidth.

  Note: Ceiling value is set using the symqos -rcopy command.

- List, query, and verify copy sessions to display the current session status.
- Export the run information to an output file.

For arrays running Enginuity 5876 or lower, the following optional settings and actions are available for arrays running Enginuity 5876 or lower:

- Data protection and recovery options for hot pulls.
- Enable or disable front-end zero detection for pull operations to thin control devices.
- Background copying mode of a session.
- Ceiling value for bandwidth.

  Note: Ceiling value is set using the symqos -rcopy command.

- Session pace for copying, recreating, and restoring.
- Differential copying.
- List, query, and verify copy sessions to display the current session status.
- Remove a remote device from a session.
- Recreate a differential copy session.
- Rename a differential copy session.
- Restore data from a remote device of a copy session.
- Export the run information to an output file.

Note: For detailed syntax of the `symrcopy` command, refer to the *EMC Solutions Enabler CLI Command Reference*.

Creating a device file

Before an Open Replicator copy session can be created, a device file must be created that lists the control and remote device pairs for the copy operation. The device file syntax contains two columns (for control and remote devices). Devices in the file must be specified either by their unique LUN WWN, or by the storage array ID and device number (`Storage ID:device#`). Valid identifiers for devices are `WWN`, `symdev` (array device).

Use the following rules to determine the correct device ID format in the device file:

- If the array for the remote device is visible to the host where the `symrcopy` command is run (locally or by a remote RDF connection), either the storage array ID and device number or the LUN WWN can be used for the remote device ID in the device file.
- If the array for the remote device is only visible using the `symsan` command, only the LUN WWN can be used for the remote device ID in the device file.

Obtaining device information

The Solutions Enabler (SYMCLI) provides several commands that can be used to obtain device information, including device number, director information, WWN, and capacity. This information is helpful in determining and identifying devices for inclusion in the device file. Some of these commands include: `symdev`, `syminq`, `sympd`, `symsan` and `symstat`. Example usage for `symdev` and `syminq` are provided below. For detailed information on using these and other SYMCLI array management commands, search for these command names in this document.

Device information for a given array is obtained by using the `symdev list` command with the array ID (`-sid`) option. For example, for a list of devices on array 041, enter:

```
symdev list -sid 0041
```

```
Symmetrix ID: 000187900041

        Device Name          Directors                   Device
--------------------------- ------------ -------------------------------------
                                                                          Cap
Sym   Physical                SA :P DA :IT  Config        Attribute    Sts   (MB)
--------------------------- ------------- -------------------------------------

00102 /dev/vx/rdmp/c5t0d2s2  03A:0 01A:C2  2-Way Mir     Grp'd    (M) RW   17261
00103 /dev/vx/rdmp/c5t0d3s2  03A:0 01D:C3  2-Way Mir     Grp'd    (M) RW   17261
00104 /dev/vx/rdmp/c5t0d4s2  03A:0 16C:D2  TDEV          N/Grp'd  (M) RW   17261
00105 /dev/vx/rdmp/c5t0d5s2  03A:0 16C:C3  TDEV          N/Grp'd  (M) RW   17261
00106 /dev/vx/rdmp/c5t0d6s2  03A:0 01A:C4  RAID-5        Grp'd    (M) RW   17261
00107 /dev/vx/rdmp/c5t0d7s2  03A:0 01A:D5  RAID-5        Grp'd    (M) RW   17261
<. . .>
```

Note: The output in the above example has been truncated.

Note: For details about `symdev` command syntax, refer to the *EMC Solutions Enabler CLI Command Reference*.

The following example shows a SCSI inquiry of all devices listed by the device WWN:

```
symiq -sym -wwn

Device                                              Device
------------------------------ ---------------- --------------------------------
Name                        Num    Array ID         WWN
------------------------------ ---------------- --------------------------------
/dev/vx/rdmp/c5t0d2s2      00168 000000006190     60060480000190300016533030303238
/dev/vx/rdmp/c5t0d3s2      001F8 000000006190     60060480000190300016533030303239
/dev/vx/rdmp/c5t0d4s2      001F9 000000006190     60060480000190300016533030303241
/dev/vx/rdmp/c5t0d5s2      00170 000000006190     60060480000190300016533030303242
/dev/vx/rdmp/c5t0d6s2      00172 000000006190     60060480000190300016533030303243
/dev/vx/rdmp/c5t0d7s2      001B2 000000006190     60060480000190300016533030303244
```

Note: For details about `syminq` command syntax, refer to the *EMC Solutions Enabler CLI Command Reference*.

Creating the file

The control device is always listed in the first column of the device file. Lines in the device file that begin with a pound symbol (#) will be ignored. The device filename (`-file Filename`) is inserted into the command line for control operations. The device text file shown below lists control and remote devices for six copy sessions. Each line in the device file is a separate copy session.

```
# dev_file_1
## column1:control column2:remote
# Symmetrix and StorageID:device always listed first
symdev=000000006190:0168 wwn=60060480000000000619053594D314638
symdev=000000006190:01F8 wwn=60060480000000000619053594D314640
symdev=000000006190:01F9 wwn=60060480000000000619053594D314637
symdev=000000006190:0170 wwn=60060480000000000619053594D314642
symdev=000000006190:0172 wwn=60060480000000000619053594D314646
symdev=000000006190:01B2 wwn=60060480000000000619053594D314649
# End
```

The following file will control a session with one control device (01) and multiple remote devices (41 and 42). This is only used with cold push sessions:

```
symdev=000000001234:01 symdev=000000005678:41
symdev=000000001234:01 symdev=000000005678:42
```

The following file shows a mix of symdev and wwn device ID usage:

```
symdev=000000001234:01 symdev=000000005678:42
symdev=000000001234:02 symdev=000000005678:43
symdev=000000001234:03 wwn=60060480000000000567853594D303434
```

Exporting device runs to a file

A session device list can be exported to a text file.

The following example shows how to use the `symrcopy export` command to create a file containing a list of device runs corresponding to a session name:

```
symrcopy export -session_name rcopy_1 -file dev_file_1.txt
```

The output file (`dev_file_1.txt`) contains the session device list, as shown here:

```
symdev=000000006190:0168 wwn=60060480000000000619053594D314638
symdev=000000006190:01F8 wwn=60060480000000000619053594D314640
symdev=000000006190:01F9 wwn=60060480000000000619053594D314637
```

```
symdev=000000006190:0170 wwn=60060480000000000619053594D314642
symdev=000000006190:0172 wwn=60060480000000000619053594D314646
symdev=000000006190:01B2 wwn=60060480000000000619053594D314649
```

Creating a session

An Open Replicator copy session is created by using the `symrcopy create` command to define a new session. Other mandatory syntax session controls that are included in the `symrcopy create` command line are the copy direction parameter (`-push|-pull`), the online/offline parameter (`-hot|-cold`), and the device text filename (`-file` *`Filename`*).

These parameters identify the direction for the copy operation, what directors will be used, and the filename of the device text file providing the list of the control and remote devices. Refer to "Creating a device file" on page 601 for instructions on how to create the device file.

Note: A session name (`-name`), which can be used for control operations, can also be included in the `symrcopy create` command.

When creating a copy session, the `symrcopy` command must indicate whether the Symmetrix control devices will be pushing data to, or pulling data from the remote devices in the copy session. A pull operation signifies that data will be copied through the SAN from remote devices to a Symmetrix control devices. A push operation signifies that data will be copied from Symmetrix control devices to remote devices within the SAN.

The copy direction for a session is designated by a flag in the output of `symrcopy list` command. Refer to "Monitoring session status" on page 611 for more detail.

The following example shows how to define an Open Replicator copy session using the `symrcopy` command:

```
symrcopy create -name rcopy_1 -pull -hot -file dev_file_1
```

In the above example, the name of the copy session is assigned as `rcopy_1`. The copy session is identified as a pull (`-pull`) operation, so data will be copied to the control devices from the remote devices. Devices will be copied online (`-hot`), meaning that all directors mapped to the local devices will be used for the copy operation.

Note: Arrays running HYPERMAX OS 5977 dynamically find directors and ports that can access an ORS session's remote devices, and use these directors and ports for data transfer.

After the session is activated, all device tracks are background copied in the CopyInProg state. The name of the device text file to be used for the session is specified as `dev_file_1`.

Precopying

For hot push operations only, the `-precopy` option is used with the `create` or `recreate` command to begin copying session data immediately in the background before the session is activated. The `-precopy` option does not need to be specified in the `symrcopy create` command.

The following example shows how to define an Open Replicator copy session using the `-precopy option`.

```
symrcopy create -name rcopy_1 -precopy -file dev_file_1
```

Refer to "Background copying" on page 609 for more information on background copying mode options.

CAUTION

Potential data loss could occur during a hot pull operation in the event of a SAN failure or other connectivity issue. For optional data protection against such failures, refer to "Donor update" on page 607.

Note: For pull operations from devices with SCSI reservations, if the remote devices have a cluster running against them or the devices are AIX LVM devices, the cluster, AIX host, or other software that is creating the SCSI reservations must be shut down before creating the Open Replicator session.

Differential copying

Differential copying is the default for a push operation only, so the `-differential` option does not need to be specified in the `symrcopy create` command. Differential copying allows sessions to be to recreated at a later time using the `symrcopy recreate` command. The `recreate` command is used to incrementally update from the control to remote devices. Refer to "Recreating a session" on page 618 for more information.

Differential copying also allows sessions for a push operation to be restored back to the control device by using the `symrcopy restore` command. Refer to "Restoring a session" on page 619 for more information.

With the `-nodifferential` option specified in the `symrcopy create` command, sessions cannot be recreated or restored.

The `-nodifferential` option is the default for pull operations.

The differential copy status for a session is designated by a flag in the output of `symrcopy list` command. Refer to "Monitoring session status" on page 611 for more detail.

Hot and cold copying

Open Replicator copy sessions running *online* (`-hot`) are up and running other host applications, such as a database or file server, while devices are being copied. All directors mapped to the control host must be able to copy to the remote devices.

For Enginuity 5876 or lower, there is a limit of 32 directors mapped to a device that is involved in an ORS hot session. If a device is mapped to more then 32 directors, the session creation fails.

Note: Arrays running HYPERMAX OS 5977 dynamically find directors and ports that can access an ORS session's remote devices, and use these directors and ports for data transfer.

A hot copy session that is copying data from the host control array to other devices located in the SAN (`-push`) are limited to a single control device copying to a single remote device. The control device may be read/write enabled to the host, but the remote device should not be accessible from its host.

During hot push operations, if the control array attempts to write to control device tracks that have not yet been copied, Open Replicator immediately copies those tracks to the remote device first and then services the I/O request.

⚠CAUTION

Potential data loss could occur during a hot pull operation in the event of a SAN failure or other connectivity issue. For optional data protection against such failures, refer to **"Donor update" on page 607****.**

With an *offline* (`-cold`) copy session, directors are used that are mapped to the control device and can reach the remote device. The control device must be set as user Not Ready to the array host. Concurrent data copying from one control device to up to 16 remote devices, is only allowed with a cold push operation. Refer to Figure 26 on page 606 for a depiction of a cold copy session to multiple remote devices.

Note: For arrays running HYPERMAX OS 5977, only one remote target is supported.

When the `symrcopy create` command is executed, it initiates a discovery process, which enables the directors that are mapped to the control devices to discover the remote devices for the session. With hot copying, all directors that have the control devices mapped are required to participate in the session. If any director cannot see the remote device after the discovery process, the session creation will fail. This situation can be avoided with cold copying, as the session will be created as long as one or more directors have discovered the remote device during the discovery.

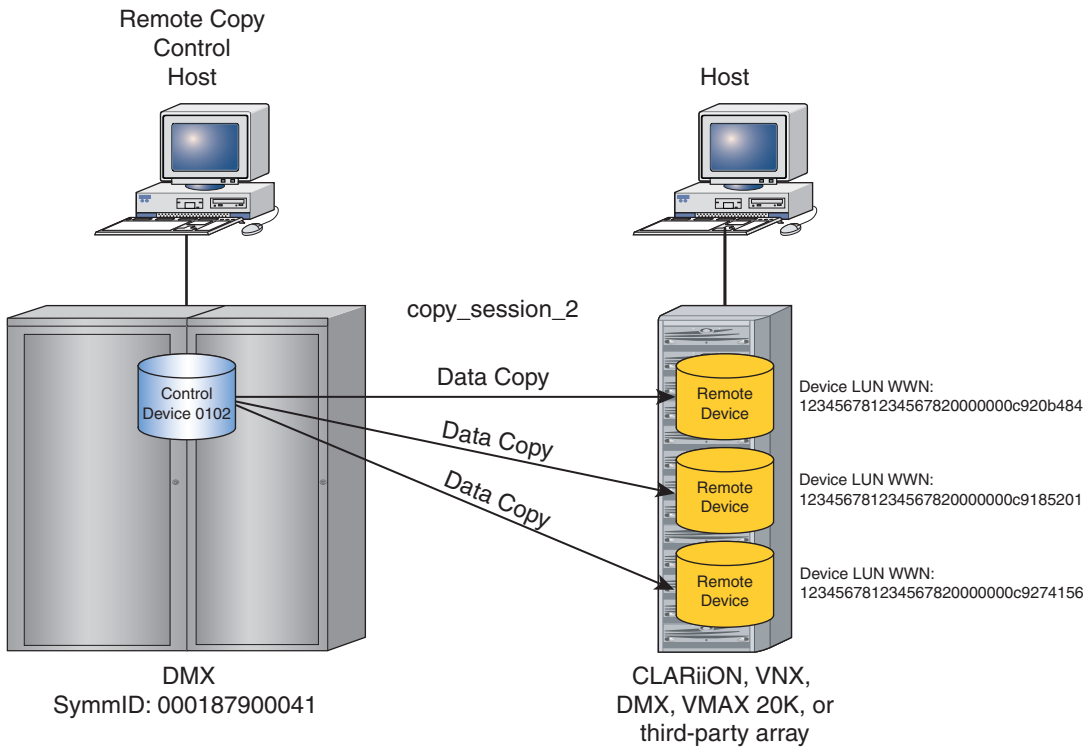
Note: Arrays running HYPERMAX OS 5977 dynamically find directors and ports that can access an ORS session's remote devices, and use these directors and ports for data transfer.

If there is a network failure while hot copying from a control array to other devices in the SAN (`-push`), the copy session will fail. However, the local host can still continue writing to the control device without impact to the application. If there is a network failure while performing a cold pull operation, the session will stall and retry.

With cold copying, the control device is already disabled for both reads and writes (Not Ready) to the control host. Open Replicator will continue to attempt the copy operation until the network is operational again. For the `symrcopy create` operation to be successful, devices must be in the proper state when the command is initiated.

Note: Failed push sessions can be activated again only if no new data has been written to the control device since the session failed. Refer to "Recovering from a failed session" on page 618 for more information.

The online or offline status for a session is designated by a flag in the output of `symrcopy list` command. Refer to "Monitoring session status" on page 611 for more detail.




```
symrcopy create -name copy_session_2 -push -cold -file multi_tgt
symrcopy activate -file multi_tgt
```

SYM-001471

Figure 26 Cold push copy session: single device to multiple remote devices

Front-end zero detection

Front-end zero detection is an option which can be used with thin control devices. Front-end zero detection looks for incoming zero patterns from the remote device, and instead of writing the incoming data of all zeros to the thin control device, the group on the thin device is de-allocated. Front-end zero detection is allowed for `-pull` operations only and is indicated by the `-frontend_zero` option. This option is ignored for any standard control devices.

The front-end zero detections commands are ignored for any standard control devices.

Note: With front-end zero detection enabled, "persistent" allocations will be treated as regular allocations, and track group will be de-allocated.

Enabling front-end zero detection

By default pull sessions have `-frontend_zero` disabled. The following example shows how to enable front-end zero detection:

```
symrcopy create -session_name rcopy_1 -pull -hot -frontend_zero -file
dev_file_1
```

Disabling front-end zero detection

The `set frontend_zero off` command disables zero detection for the session, and is only allowed during active `-pull` operations.

The following example shows how to disable front-end zero detection:

```
symrcopy -session_name rcopy_1 set frontend_zero off
```

The front-end zero detection status for a session is designated by a flag in the output of `symrcopy list` command. Refer to "Monitoring session status" on page 611 for more detail.

Note: Once front-end zero detection is disabled during an active session, it cannot be enabled again during the session.

Donor update

To protect against potential data loss due to a SAN failure or other connectivity issue during a hot pull operation, use the `-donor_update` option. With this option, all writes to the control device from the host are immediately copied to the remote device as well. Because the data is fully copied to both the remote device and the control device, if a failure occurs, the session can be safely terminated and created again to fully recover from any mid-copy failure.

Enabling donor update option

The following example shows how to create and activate an Open Replicator copy session for a hot pull operation using the `donor_update` option:

```
symrcopy create -name rcopy_1 -pull -hot -donor_update -file
dev_file_1
symrcopy activate -session_name rcopy_1
```

Note: For information on the `activate` command, refer to "Activating a session" on page 608.

If during an activated hot pull operation, a SAN failure or other connectivity issue is detected, then terminate the Open Replicator sessions.

The following example shows how to terminate the Open Replicator sessions associated with the control device:

```
symrcopy terminate -file dev_file_1 -symforce
```

The following example shows how to start the copy session again after the problem has been resolved:

```
symrcopy create -name rcopy_1 -pull -hot -donor_update -file
dev_file_1
symrcopy activate -session_name rcopy_1
```

Note: The above example restarts the copy process from where it left off at the time of failure.

The donor update status for a session is designated by a flag in the output of `symrcopy list` command. Refer to "Monitoring session status" on page 611 for more detail.

Note: The `donor_update` option must be included in the original `symrcopy create` command to fully recover all writes made to the devices prior to the failure.

Disabling donor update option

The donor update option may be disabled by using the `set donor_update off` command. This command stops the copying of data to the remote devices. Thus stopping all new writes to the control device from being immediately copied to the remote device. When used with the `-consistent` option, the consistency of the data on the remote devices will be maintained. When used without the `-consistent` option, donor update will still be deactivated, but consistency on the remote devices will not be maintained.

Note: The `set donor_update off` command requires Enginuity version 5773 or higher.

This option is useful for a hot pull session, with donor update enabled, where the session has finished copying and maintaining a consistent image on the remote devices is desired. By using the `set donor_update off` command with the `-consistent` option after the session has fully copied, the donor update portion of the session is disabled, but data consistency on the remote devices is maintained.

The following example shows how to set the donor update option to off and maintain consistency on the remote devices:

```
symrcopy set donor_update off -session_name rcopy_1 -consistent
```

Note: The `donor_update` option may also be turned off while the session is in the CopyInProg (copy in progress) state by including the `-force` option in the command line. The session will continue to copy in its current mode without donor update.

If the session is terminated, renamed, restored, recreated, a device is removed, or another session is created using the same control device, the donor update portion of the session is automatically deactivated and consistency on the remote devices is lost. To maintain the consistency on the remote devices, issue the `set donor_update off -consistent` command prior to any of these actions.

Note: The donor update option may also be used on an incremental restore session. Refer to "Restoring a session" on page 619 for more information.

Activating a session

To begin the copying process for an Open Replicator copy session, use the `symrcopy activate` command and specify the device file (`-file` *`Filename`*) or the session name (`-session_name` *`SessionName`*). This activates the copy sessions for device pairs listed in the device file and begins copying data to (*pushing*) or from (*pulling*) the remote devices.

The following example shows how to activate an Open Replicator copy session using the `symrcopy` command:

```
symrcopy activate -session_name rcopy_1
```

The above example starts the copy session that was previously created in the example shown in "Creating a session" on page 603.

Note: Any other Open Replicator copy sessions that were previously created using the specified device file (and session name) will also be started.

If control devices are pushing data to remote devices and the control devices are currently online for host I/O operations, include the Enginuity Consistency Assist (ECA) option (`-consistent`) in the command line to temporarily prevent host I/O while the Open Replicator copy session begins. This begins a consistent point-in-time copy to the remote devices using an ECA window, which temporarily freezes host I/O to the control devices.

Note: Under certain circumstances, failed sessions may be reactivated. Refer to "Recovering from a failed session" on page 618.

Background copying

Open Replicator copy sessions that are actively background copying to devices are in the CopyInProg state. This is the default state for copy sessions. This state can be changed to the CopyOnAccess state for a pull operation, the CopyOnWrite state for a push operation, or the Precopy state for a hot push operation by using the `symrcopy set mode [-copy|-nocopy|-precopy]` options. An activated session in the CopyOnAccess state copies data to the control device only when those tracks have been accessed on the control device. An activated session in the CopyOnWrite state copies data to the remote device only when those tracks are accessed on the control device.

A hot push session that is in the Precopy state will immediately begin copying data in the background before the session is activated. Session data will continuing copying to the remote device until either the mode is changed to `nocopy`, `copy`, or the session is activated, at which time a point-in-time copy of the control device is made. After the session has been activated, copying will continue in the CopyOnWrite (`nocopy`) or CopyInProg (`copy`) state. The Precopy feature is available only for hot push operations. Hot push sessions can also be set to Precopy mode by including the `-precopy` option with either the `create` or `recreate` command.

The background copy status for a session is designated by a flag in the output of `symrcopy list` command. Refer to "Monitoring session status" on page 611 for more detail.

Note: The `-precopy` option requires Enginuity version 5772 through 5876.

The following example shows how to temporarily stop the background copying for a session by changing the state to CopyOnAccess or CopyOnWrite from CopyInProg using the `symrcopy` command:

```
symrcopy set mode nocopy -file dev_file_1
```

The following example shows how to resume background copying for a session and change to the CopyInProg state:

```
symrcopy set mode copy -file dev_file_1
```

The following are examples of how to immediately begin background copying on a hot push session without making a point-in-time copy:

```
symrcopy set mode precopy -file dev_file_1
```

or

```
symrcopy create dev_file_1 -precopy
```

Note: Precopy mode can only be set when the session is not activated.

Note: To see an example of the `-precopy` option as used with the recreate command, refer to "Recreating a session" on page 618.

Setting the ceiling value

Ceiling values can be adjusted to optimize the performance of the specific SAN environment.

The `symqos -rcopy set ceiling` command sets the maximum allowed bandwidth percentage for a given director, port, director/port pair, or all directors and ports. Valid values are 0 - 100 (%), DISABLE or NONE (shuts off the ceiling function).

Note: The DISABLE setting is only allowed for HYPERMAX OS 5977 or higher.

The following example shows how to set a bandwidth ceiling of 100% for all directors on array 6190:

```
symqos -rcopy set ceiling 100 -dir all -sid 6190
```

The new ceiling setting is viewed by using the following `list ceiling` command:

```
symqos -rcopy list ceiling -dir all -sid 6190
```

For example, setting the ceiling value to 100% causes Open Replicator to consume as much bandwidth as possible, typically:

- 80 MB/s for a 1 GB SAN
- 150 MB/s for a 2 GB SAN
- 180 MB/s for a 4 GB SAN (for DMX - 4 systems)
- 300 MB/s for a 4 GB SAN (for VMAX Family arrays)
- 300 MB/s for an 8 GB SAN (for VMAX Family arrays)

Setting the ceiling to a value (other than NONE) renders the session pace value ineffective to the copy. If the ceiling value is set to NONE, the session pace is in effect for the copy.

When using ORS with SRDF/A, users should monitor and adjust the ORS ceiling/session pace or turn on SRDF/A pacing to prevent ORS I/O from causing SRDF/A to drop. Refer to Primus case emc292509 for guidelines on setting these values.

Setting the session pace

If the ceiling value is set to NONE, the session pace can be set for devices being copied, recreated, or restored to manage the speed of the replication process. The session pace designates how fast data copies between devices. Values can range from 0 to 9, with 0 being the fastest pace, and 9 being the slowest pace. If set to 0, there is no inserted delay time and the replication will proceed as fast as possible.

Values of 1 - 9 add delays, which takes longer to complete copying but conserves system resources. The default for both online (hot) replication and offline (cold) replication is 5.

The following example shows how to set the session copy pace:

```
symrcopy set pace 0 -file dev_file_1
```

Note: The session pace becomes ineffective to the copy if the ceiling is set to a value other than NONE.

Note: For arrays running HYPERMAX OS 5977 or higher, setting session pace is not supported.

Monitoring session status

Open Replicator session status is checked using the `symrcopy query`, `symrcopy list`, or `symrcopy verify` command.

Listing all sessions

To list the Open Replicator copy sessions for a local array, use the `symrcopy list` command. This command returns status information for all created sessions. To list session information for a specific Symmetrix array, include the array ID (`-sid` *SymmID*) option in the command line.

The following options are available for the `symrcopy list` command:

- `-offline` option displays only information held in the database and does not query the array for updated session information.
- `-detail` option displays additional device information for modified tracks, session pace, and session name.
- `-wwn` option displays the full device world wide name.

Note: Using the `-detail` and `-wwn` options expands the width of the character display, which may not view properly for some displays.

The following is a list example of all Open Replicator sessions for array ID 6190:

```
symrcopy list -sid 6190
```

```
Symmetrix ID: 0000000006190

     Control Device              Remote Device                Flags        Status       Done
    ----------------------- -------------------------------- -------      --------     -----
                   Protected
         SID:symdev    Tracks    Identification              RI  CDSHUTZ SRC <=> TGT      (%)
```

```
------------  --------- -------------------------------- -- ------- -------------   ----
   00168         33000 60060480000000000619053594D314638 .W X..XXM. Copied           100
   001F8         33000 60060480000000000619053594D314640 .W X..XXS. CreateInProg     N/A
   001F9         33000 60060480000000000619053594D314637 .W X..XXS. CreateInProg     N/A
   00170         20000 60060480000000000619053594D314642 .W X..XXSX CopyInProg        50
   00172         30000 60060480000000000619053594D314646 .W X..XXSX CopyInProg        75
   001B2         33000 60060480000000000619053594D314649 .W X..XXRX CopyInProg        75

   Total      ---------
    Tracks     152000
    MB(s)       12062.5

  Legend:
   R (Remote Device Vendor Identification)
   S = Symmetrix, . = Unknown.

   I:  (Remote Device Specification Identifier)
     D = Device Name, W = LUN WWN, World Wide Name.

   Flags:
   (C): X = The background copy setting is active for this pair.
        . = The background copy setting is not active for this pair.
   (D): X = The session is a differential copy session.
        . = The session is not a differential copy session.
   (S): X = The session is pushing data to the remote device(s).
        . = The session is pulling data from the remote device(s).
   (H): X = The session is a hot copy session.
        . = The session is a cold copy session.
   (U): X = The session has donor update enabled.
        . = The session does not have donor update enabled.
   (T): M = The session is a migration session.
        R = The session is a RecoverPoint session.
        S = The session is a standard ORS session.
   (Z): X = The session has front-end zero detection enabled.
        . = The session does not have front-end zero detection enabled.
   (*): The failed session can be reactivated.
```

Filtering session types

The list `symrcopy list` command provides the `-type` option that lists either standard ORS sessions or RecoverPoint sessions.

Note: For listing Federated Live Migration (FLM) sessions, refer to "Listing a FLM session" on page 689.

The following is a list example of standard sessions only:

```
symrcopy list -sid 6190 -type standard
```

```
Symmetrix ID: 0000000006190

   Control Device              Remote Device                 Flags    Status       Done
   ----------------------- ------------------------------- -------  --------     -----
                Protected
   SID:symdev   Tracks     Identification                  RI CDSHUTZ SRC <=> TGT   (%)
   ----------- --------- ------------------------------- -- ------- ------------  ----
   001F8            33000 60060480000000000619053594D314640 .W X..XXS. CreateInProg  N/A
   001F9            33000 60060480000000000619053594D314637 .W X..XXS. CreateInProg  N/A
   00170            20000 60060480000000000619053594D314642 .W X..XXSX CopyInProg     50
   00172            30000 60060480000000000619053594D314646 .W X..XXSX CopyInProg     75

   Total        ---------
    Tracks        152000
    MB(s)         12062.5
```

The following is a list example of RecoverPoint sessions only:

```
symrcopy list -sid 90 -type recoverpoint
```

```
Symmetrix ID: 000000006190

         Control Device            Remote Device                  Flags      Status      Done
         --------------- ------------------------------------- --------    ------     -----
                      Protected
     SID:symdev       Tracks    Identification                  RI CDSHUTZ SRC <=> TGT    (%)
    ------------    --------- --------------------------------  -- ------ -------------- ----
     01B2                 33000 60060480000000006190535 94D314649  SW ...XXR. Created        75

        Total        ---------
         Tracks          33000
         MB(s)            2062.5
```

For the output legend, refer to page 612.

Note: If the `symrcopy list` command is run without the `-type` option, the default behavior is to include all sessions in the list, as shown in "Listing all sessions" on page 611.

Querying session status

Open Replicator session status is checked using the `symrcopy query` or `symrcopy verify` command.

The `symrcopy query` command is used to display details for remote copy sessions defined in a device file. The query command provides current status information for control/remote device pairs. If the device pair state is CopyInProg, the query command displays the percentage of copying that has completed.

The following options are available for the `symrcopy query` command:

- `-i` (interval) is used to execute the query command in repeated intervals (in seconds). The default for interval is 30 seconds if the count option is used, and 15 seconds is the minimum interval that can be specified.
- `-c` (count) is used with the `-i` and specifies the duration of the query intervals. See "Detailed query for copy session status" for an example command with the `-c` and `-i` options .

Note: Estimated time to copy completion is shown in the query output when the `-i` and *-c* options are used or if the protected track count has changed since the last interval.

- `-offline` option displays only information held in the database and does not query the array for updated session information.

Note: If the `-offline` option is included with the `symrcopy query` command, the `-i` and *-c* options are not useful as the output will not change between intervals.

- `-detail` option displays additional device information for modified tracks, session pace, and session name.
- `-wwn` option displays the full device world wide name.

Note: Using the `-detail` and `-wwn` options expands the width of the character display, which may not view properly for some displays.

Detailed query for copy session status

The following example shows how to query for copy session status:

```
symrcopy query -file dev_file_1
```

```
       Control Device                  Remote Device             Flags     Status       Done
---------------------------- ----------------------------------- ----- -------------    ----
                    Protected
SID:symdev          Tracks    Identification                    RI CDSHUTZ SRC <=> TGT     (%)
------------------ --------- ---------------------------------- -- ------- -----------    ----
000000006190:00168      33000 60060480000000000619053594D314638 .W X..XXM. Copied         100
000000006190:001F8      33000 60060480000000000619053594D314640 .W X..XXS. CreateInProg   N/A
000000006190:001F9      33000 60060480000000000619053594D314637 .W X..XXS. CreateInProg   N/A
000000006190:00170      20000 60060480000000000619053594D314642 .W X..XXSX CopyInProg      50
000000006190:00172      30000 60060480000000000619053594D314646 .W X..XXSX CopyInProg      75
000000006190:001B2      33000 60060480000000000619053594D314649 .W X..XXRX CopyInProg      75

Total               ---------
  Tracks               152000
  MB(s)                12062.5
```

For the output legend, refer to page 612.

The following example shows a query command that will be run every 30 seconds for 1 hour:

```
symrcopy query -file dev_file_1 -i 30 -c 120
```

The `symrcopy query` command can also be used with a session name, as shown in the following example:

```
symrcopy query -session_name rcopy_2
```

Summary query for copy session status

The output from the summary query is an abbreviated listing that shows all possible session states and the number of sessions that are in each state.

The `-wwn` and the `-detail` options cannot be used with the `-summary` option.

The following example shows how to query for a copy session status with `-summary` option:

```
symrcopy -file dev_file_1 query -summary
```

```
Device File Name : dev_file_1

  RCopy Session State           Count
  -----------------------      ------
  CreateInProg                      2
  Created                           0
  RecreateInProg                    0
  Recreated                         0
  CopyInProg                        3
  CopyOnAccess                      0
  CopyOnWrite                       0
  Copied                            1
  SyncInProg                        0
  Synchronized                      0
```

```
Restored                          0
RestoreInProg                     0
Precopy                           0
TerminateInProg                   0
Failed                            0
Stopped                           0
FailedBack                        0
VerifyInProg                      0
Invalid                           0
-----------------------      -------
Total                             6

                                Track(s)    MB(s)
                             -----------  -------
Total Protected                  152000   12062.5
```

Verifying session state

The `symrcopy verify` command can be used to verify copy session states. Command syntax for `verify` is as follows:

```
symrcopy verify [-createinprog | -created | -recreateinprog |
-recreated | -copyinprog | -copyonaccess | -copyonwrite | -copied |
-terminateinprog | -failed | -verifyinprog | -restored |
-restinprog | -precopy [-cycled] | -syncinprog | -synchronized |
-failedback]
```

Table 41 Verifying session state (page 1 of 2)

Verify option	Description
`-createinprog`	Verifies that the copy session is in the process of being created.
`-created`	Verifies that the copy session has been created.
`-recreateinprog`	Verifies that the copy session is in the process of being recreated (incrementally updating the targets).
`-recreated`	Verifies that the copy session has been recreated. Device pairs in the session have finished incrementally updating.
`-copyinprog`	Verifies which device pairs in the copy session are currently in the CopyInProg state (actively background copying).
`-copyonaccess`	Verifies which device pairs in the copy session are currently in the CopyOnAccess state (only copying the device tracks to the control device as they are being accessed on the remote device for a pull operation).
`-copyonwrite`	Verifies which device pairs in the copy session are currently in the CopyOnWrite state (only copying the device tracks to the remote device as they are being written to on the remote device for a push operation).
`-copied`	Verifies which device pairs in the copy session have finished copying data. This is the default if no option is provided.
`-terminateinprog`	Verifies that the copy session is in the process of terminating.
`-failed`	Verifies if any of the device pairs in the copy session have failed to copy.
`-failedback`	Verifies if a FLM session has failed back.
`-verifyinprog`	Verifies that all active directors for the copy session have completed copy operations.

Table 41 Verifying session state (page 2 of 2)

Verify option	Description
-precopy	Verifies that the device pair is currently in the Precopy state (copying device tracks in the background without activation). Adding the -cycled option verifies all precopy sessions that have completed one cycle.
-restinprog	Verifies that the copy session is in the process of being restored.
-restored	Verifies that the copy session has been fully restored.
-syncinprog	Verifies which device pairs in the copy session are currently in the SyncInProg state (actively background copying).
-synchronized	Verifies which device pairs in the copy session are in the Synchronized state.

The following example shows the output result for the verify command:

```
symrcopy -file dev_file_1 verify

One of the device(s) in the list are in 'Copied' state.
```

The following example shows the output result when the verify command is used with the -summary option:

```
symrcopy -file dev_file_1 verify -summary
```

Note: The one-line verify command output comes after the -summary output.

```
Device File Name : dev_file_1

RCopy Session State           Count
  -----------------------      ------
  CreateInProg                      2
  Created                           0
  RecreateInProg                    0
  Recreated                         0
  CopyInProg                        3
  CopyOnAccess                      0
  CopyOnWrite                       0
  Copied                            1
  SyncInProg                        0
  Synchronized                      0
  Restored                          0
  RestoreInProg                     0
  Precopy                           0
  TerminateInProg                   0
  Failed                            0
  Stopped                           0
  FailedBack                        0
  VerifyInProg                      0
  Invalid                           0
  ------------------------      ------
  Total                             6
```

```
                                   Track(s)    MB(s)
                                -----------  -------
Total Protected                   156000    12062.5

One of the device(s) in the list are in 'Copied' state.
```

Note: If no verify option is provided, then “copied” is the default state that is verified.

Terminating a session

To terminate a copy session and remove it from the array, use the `symrcopy terminate` command. If the session is activated and in the CopyInProg, CopyOnAccess, or CopyOnWrite state, the `-symforce` option is mandatory to terminate the session.

The following are examples of how to terminate a copy session that has finished copying or has not yet been activated using the `symrcopy` command:

```
symrcopy terminate -file dev_file_1
```

or

```
symrcopy terminate -session_name rcopy_1
```

The following are examples of how to terminate a copy session that has been activated and has not yet finished copying:

```
symrcopy terminate -file dev_file_1 -symforce
```

The following example shows how to terminate all sessions associated with the control device:

```
symrcopy terminate -all_sessions -symforce -file dev_file_1
```

Note: Remote devices in the device file are ignored.

CAUTION

Use care when applying the `-symforce` option to terminate an active session. At termination, the receiving devices will contain an incomplete data copy and should be considered invalid.

Terminating a RecoverPoint session

The `symrcopy` command does not support control of Open Replicator RecoverPoint sessions, however for cleanup purposes RecoverPoint sessions can be terminated and removed as follows:

```
symrcopy terminate -file dev_file_1 -rp
```

Removing a remote device from a session

To remove a remote device from a session, it must be put in the device file.

The following example shows the `symrcopy remove` command:

```
symrcopy remove -file dev_file_1
```

Note: The remote device must be from a differential session in the Copied state.

Recreating a session

For differential push operations only, the copy session can be recreated using the `symrcopy recreate` command. The session must have been originally created with differential copying. Activating a recreated session begins an incremental update of the devices to copy any device tracks that were changed since the last time the copy session actively finished copying. Up to 15 sessions can be created for incremental updates per logical volume. Open Replicator uses the Symmetrix Differential Data Facility (SDDF) to set the track protection bitmaps and monitor track differences between the control and remote devices.

Note: Recreating a session creates a new point-in-time copy of the data.

The following example shows how to recreate and activate a copy session for incremental track updates:

```
symrcopy recreate -name rcopy_2 -file dev_file_3
symrcopy activate -session_name rcopy_2
```

When a session is recreated, the session can be renamed using the `-name` option. For hot push operations, a copy session can be recreated to pre copy the incremental track updates in the background without activating the session by including the `-precopy` option in the command line. This shown in the following example:

```
symrcopy recreate -name rcopy_2 -file dev_file_3 -precopy
```

Note: The `-pace` option can be included in the command line to manage the speed of the replication process. Refer to "Setting the session pace" on page 611.

Recovering from a failed session

Failed sessions can be reactivated by issuing the `activate` command again, only if there has been no new data written to the control device since the failure occurred. If new data is indicated on devices that are part of the session, session activation will be blocked. If session activation is blocked and it is a non-differential session, terminate the session and issue the `create` and `activate` commands again, which will begin the copy from the beginning. If session activation is blocked and it is a differential session, recreate the session to create a new point-in-time copy.

Note: Support for reactivating failed sessions requires Enginuity version 5773 or higher.

When no new data is indicated on the devices in the session, the session is eligible to be reactivated. Use the `symrcopy query` command to check the status of a failed session. Failed sessions eligible for reactivation are indicated by a failed(*) status that includes an asterisk (*) symbol.

The following example presents a query output showing the failed devices in the session. The sessions shown are eligible for reactivation.

```
symrcopy -file dev_file_1 query -detail
```

```
Symmetrix ID: 000190300237

     Control Device                Remote Device            Flags     Status     Done Pace Name
------------------------- -------------------------------- ----- -------------- ---- --- ----
      Protected Modified
Sym   Tracks    Tracks    Identification                RI   CDSHUTZ CTL<=>  REM (%)
----- --------- --------- ------------------------------ -- -------- ------------ ---- --- ---
001F8   33000         0  006048000000000619053594D314640 SD  X.XX.S.   Failed (*) N/A 5   N/A
001F9   33000         0  006048000000000619053594D314637 SD  X.XX.S.   Failed (*) N/A 5   N/A

Total ---------
 Tracks   66000
 MB(s)     4062.4
```

For the output legend, refer to page 612.

Restoring a session

For differential push operations only, the copy session is restored back to the control device by pulling back only the changed tracks from the remote device. The session must have been created with differential copying, and must be in the copied state. Hot or cold differential push sessions can be restored.

Note: Refer to the *EMC Solutions Enabler Installation Guide* for license information required for restore functionality.

For example, if all data is copied from the control device to the remote device(s) and then changes are made to the control device, use the `symrcopy restore` command to recover the original data from the remote device. When the command is issued, the session is recreated in restore mode and automatically activated. At the start of the restore operation, all control devices will be set to Not Ready status. If running a hot session, control devices will be returned to Ready status at the end of the operation (as the data begins copying). If running a cold session, the control devices will remain in Not Ready status.

The following example shows how to restore original data from a differential push session back to the control device:

```
symrcopy restore -file dev_file_3
```

Note: A session name can also be provided by using the `-session_name` option.

Note: The `-pace` option can be included in the command line to manage the speed of the replication process. Refer to "Setting the session pace" on page 611.

Restoring data using donor update

Differential push operations may be restored using the `-donor_update` option. Using this option with the `symrcopy restore` command, a copy is maintained on the remote device of any new data that has been written to the control device while the session is in the process of restoring.

The following example shows how to restore data back to the control device using the donor update option:

```
symrcopy restore -file dev_file_3 -donor_update
```

Note: The control device will be set to not ready before the operation and then set back to its previous state after the restore has begun.

If `-donor_update` option is used, a session cannot be renamed or devices cannot be removed from a session that is in the Restored state. The `-force` option must be used to create a new session using the same control device, recreate, or restore sessions that are in the Restored state when donor update is enabled. The `-force` option must be used to terminate a session that has donor update enabled.

After the session has finished restoring to the control device, use the `set donor_update off -consistent` command to deactivate the donor update portion of the session and maintain the consistency of data on the remote devices.

The following is an example how to set the donor update option to off and maintain consistency on the remote devices:

```
symrcopy set donor_update off -file dev_file_3 -consistent
```

Note: The `-donor_update` option may also be turned off while the session is in the RestInProg (restore in progress) state by including the `-force` option in the command line. The session will continue to restore in its current mode without donor update.

If the session is terminated, renamed, recreated, a device is removed, or another session is created using the same control device, the donor update portion of the session will automatically be deactivated and consistency on the remote devices will be lost. To maintain the consistency on the remote devices, issue the `set donor_update off -consistent` command prior to any of the previously listed actions.

Using VDEVs in Open Replicator

Open Replicator supports copy operations for a cold push session from a virtual device (VDEV). To accomplish this, Open Replicator works in conjunction with TimeFinder/Snap operations by copying data from an existing Snap session. This feature provides an alternative to hot copying, where all mapped directors must participate in the session. In a cold push session from a VDEV, the VDEV is required to be mapped to at least one director. At least one of the directors, to which the device is mapped, must be zoned to the remote device.

Note: This functionality is supported only on arrays running Enginuity version 5874 through 5876. Refer to the *EMC Solutions Enabler Installation Guide* for license information on snap devices.

Note: For arrays running HYPERMAX OS 5977 or higher, creating ORS sessions on control devices that are involved in a TimeFinder session is not supported.

To begin, a snapshot of the standard device data is created on a VDEV in a TimeFinder/Snap session. The VDEV is then used as the control device in the Open Replicator copy session. Only the directors to which the VDEV are mapped are required to adhere to the zoning rules for Open Replicator. The directors to which the standard device are mapped are not required to be zoned to the remote device.

Device requirements

The following devices and configuration requirements are necessary for running an Open Replicator session using VDEVs:

- A device containing the data to be copied to the remote array.
- A VDEV to hold a snapshot of the device containing the data to be copied. The VDEV must be mapped to at least one director and at least one of the mapped directors must be zoned to the remote device.
- A remote device that is mapped to at least one director, which is zoned to the control array containing both the device that contains the data to be copied and the VDEV.

Limitations

The following limitations apply to Open Replicator sessions when using VDEVs:

- The VDEV must remain Not Ready to the user while the Open Replicator session exists.
- Only one Open Replicator session is allowed per VDEV control device. The number of Open Replicator sessions is limited to the number of VDEVs that can be created from the source device.
- Open Replicator restore to a TimeFinder/Snap session is not supported.
- Open Replicator sessions can only be created as cold differential or non-differential push operations.
- Open Replicator precopy feature is not supported.

Performing an Open Replicator session with VDEVs

The following steps explain how to run an Open Replicator session with VDEVs using TimeFinder/Snap and Open Replicator:

1. Create a TimeFinder/Snap session using the `symsnap create` command. Do not activate the session. Detailed instructions for creating a TimeFinder/Snap session and performing snap operations are provided in the *EMC Solutions Enabler TimeFinder Family CLI Product Guide*.
2. Create an Open Replicator session using the `symrcopy create` command, defining the snap device as control device. Do not activate the session. This creates an Open Replicator protected session on the VDEV.
3. Activate the Snap session using the `symsnap activate` command with the `-not_ready` option. If consistency is required, specify the `-consistent` option.
4. Activate the Open Replicator session using the `symrcopy activate` command. The Open Replicator protected session begins copying the data.
5. Wait for the Open Replicator copy session to complete. When the session has completed, the remote devices will contain a point-in-time copy of data from when the snap session was activated.

Once the Open Replicator session has finished copying all of the data, the sessions can be recreated and reactivated as many times as desired.

The following steps describe how to recreate, activate, and terminate the sessions:

1. Recreate the TimeFinder/Snap session using the `symsnap recreate` command. This command clears the VDEV pointers and sets the Open Replicator protected tracks.
2. Recreate the Open Replicator session using the `symrcopy recreate` command.
3. Activate the TimeFinder/Snap session using the `symsnap activate` command. If consistency is required, specify the `-consistent` option.
4. Activate the Open Replicator session using the `symrcopy activate` command. The Open Replicator session begins copying the data.
5. When completely done using the Open Replicator session, use the `symrcopy terminate` command to terminate the session.
6. When completely done using the TimeFinder/Snap session, use the `symsnap terminate` command to terminate the session.

"Example 5: Performing a cold push from a VDEV" on page 727 provides a detailed example.

Open Replicator examples

This section provides the following examples of using Open Replicator:

"Example 1: Performing a hot pull operation" on page 622

"Example 2: Pushing data using BCVs as the control devices" on page 637

"Example 3: Pulling data online from IBM F20 to VMAX Family array" on page 643

"Example 4: Pushing data online from VMAX Family array to Hitachi HDS 9960 array" on page 650

"Example 5: Performing a cold push from a VDEV" on page 657

Example 1: Performing a hot pull operation

This example shows how to migrate data from an older array to a VMAX Family array. The hardware setup consists of the control array whose array ID is 000187900041 (abbreviated as 41) connected to a controlling host. The remote array on the SAN is an older array (DMX). Three remote devices are each identified by their LUN WWN. Three control devices are E9, EA, and EB. The control device capacity should be equal to or larger than the remote device extents that are being copied.

Note: If a copy needs to be forced from a larger device to a smaller device (for example, data was initially copied to a larger device and now the same data needs to be copied back to the smaller device), this is done by including the `-force_copy` option with the `symrcopy create` command.

For online (`-hot`) copying, the control devices may be Read/Write enabled. The remote devices should not be receiving any updates from their local host.

The following examples and tasks show:

1. "Creating device file"
2. "Creating migration session with donor update"
3. "Querying migration session"
4. "Activating migration session"
5. "Verifying session status"
6. "Terminating migration session"

Creating device file

The first step in an ORS copy operation is to define the control/remote device pairings in a text file. A control device or remote device is specified by either its unique LUN WWN or by a combination of the array ID and the device name (array ID:device). Enter the control devices in the left-hand column, and the remote devices in the right-hand column, as shown below in the filename tango:

```
vi tango

   symdev=000187900041:E9  wwn=60060480000000000314353594D303737
   symdev=000187900041:EA  wwn=60060480000000000314353594D303738
   symdev=000187900041:EB  wwn=60060480000000000314353594D303739
```

Creating migration session with donor update

The `symrcopy create` command creates three online copy sessions so that data on the remote devices specified in file `tango` can be copied to the control devices when the copy operation is activated. The `-pull` parameter specifies that the control array is pulling the data to it. The `-hot` parameter indicates that the control array remains online during the operation. The `-name` option gives these sessions the label name `Monday`. The `-donor_update` parameter indicates that all writes to the control device from the host will also be copied to the remote device.

```
symrcopy create -name Monday -pull -hot -donor_update -file tango
-noprompt

'Create' operation execution is in progress for the device list in
device file 'tango'. Please wait...

'Create' operation successfully executed for the device list
in device file 'tango'.
```

Querying migration session

The `symrcopy query` command indicates that the sessions for the control/remote device pairs in the file `tango` are in the Created state and are considered to be active sessions. When the control host can "see" the remote devices (in this case, a remote array), Open Replicator converts the remote device LUN WWN identifier (specified in file `tango`) to the "array ID:device" format (for example, `000000003143:0077`).

```
symrcopy query -file tango

Device File Name       : tango

       Control Device                  Remote Device             Flags     Status      Done
---------------------------- ------------------------------------ ----- --------------   ---
```

```
                    Protected
SID:symdev          Tracks     Identification                   RI CDSHUTZ CTL <=> REM     (%)
------------------ --------- -------------------------------- -- ----- --------------   ---
000187900041:000E9     138090 000000003143:00077                SD X..XXS. Created        N/A
000187900041:000EA     138090 000000003143:00078                SD X..XXS. Created        N/A
000187900041:000EB     138090 000000003143:00079                SD X..XXS. Created        N/A

Total               ---------
  Track(s)             414270
  MB(s)               12945.9

        Legend:
        R (Remote Device Vendor Identification)
        S = Symmetrix, . = Unknown.

        I:  (Remote Device Specification Identifier)
        D = Device Name, W = LUN WWN, World Wide Name.

        Flags:
        (C): X = The background copy setting is active for this pair.
             . = The background copy setting is not active for this pair.
        (D): X = The session is a differential copy session.
             . = The session is not a differential copy session.
        (S): X = The session is pushing data to the remote device(s).
             . = The session is pulling data from the remote device(s).
        (H): X = The session is a hot copy session.
             . = The session is a cold copy session.
        (U): X = The session has donor update enabled.
             . = The session does not have donor update enabled.
        (T): M = The session is a migration session.
             R = The session is a RecoverPoint session.
             S = The session is a standard ORS session.
        (Z): X = The session has front-end zero detection enabled.
             . = The session does not have front-end zero detection enabled.
        (*): The failed session can be reactivated.
```

The `symrcopy activate` command activates the copy sessions for the pairings in the file `tango`. Copying from the remote array to the control array begins. At this point, migrated data on the on the control array can be accessed without waiting for the copy operation to complete.

Activating migration session

```
symrcopy activate -file tango -noprompt

'Activate' operation execution is in progress for the device list in
device file 'tango'. Please wait...

'Activate' operation successfully executed for the device list
in device file 'tango'.
```

The `symrcopy query` command with the `-detail` option indicates that the sessions for the device pairs defined in the file `tango` are in the CopyInProg state and the percent (%) completion. The display also contains other details such as the pace. The default pace value of 5 provides relatively fast copy time with only a moderate impact on the application.

```
symrcopy query -file tango -detail

Device File Name      : tango

           Control Device             Remote Device         Flags     Status Done Pace Name
------------------------------------- ------------------------------- -------- ---- ----
                  Protected Modified
```

```
SID:symdev       Tracks    Tracks     Identification      RI  CDSHUTZ  CTL <=> REM (%)
--------------- --------- --------- --------------------------------- ---------- ---- ----
000187900041:000E9  128083     0     000000003143:0077   SD  X..XXS. CopyInProg 7  5  Monday
000187900041:000EA  123742     0     000000003143:0078   SD  X..XXS. CopyInProg10  5  Monday
000187900041:000EB  127455     0     000000003143:0079   SD  X..XXS. CopyInProg 7  5  Monday

Total              ---------
  Track(s)         379280
  MB(s)           11852.5

Legend:
         R (Remote Device Vendor Identification)
         S = Symmetrix, . = Unknown.

         I:  (Remote Device Specification Identifier)
         D = Device Name, W = LUN WWN, World Wide Name.

         Flags:
         (C): X = The background copy setting is active for this pair.
              . = The background copy setting is not active for this pair.
         (D): X = The session is a differential copy session.
              . = The session is not a differential copy session.
         (S): X = The session is pushing data to the remote device(s).
              . = The session is pulling data from the remote device(s).
         (H): X = The session is a hot copy session.
              . = The session is a cold copy session.
         (U): X = The session has donor update enabled.
              . = The session does not have donor update enabled.
         (T): M = The session is a migration session.
              R = The session is a RecoverPoint session.
              S = The session is a standard ORS session.
         (Z): X = The session has front-end zero detection enabled.
              . = The session does not have front-end zero detection enabled.
         (*): The failed session can be reactivated.
```

The `symrcopy verify` command checks at 60-second intervals (`-i`) to verify whether the control/remote device pairs are in the Copied state.

Note: The Pace column will report N/A for sessions where the control device is on an array running HYPERMAX OS 5977 or higher.

Verifying session status

```
symrcopy verify -i 60 -file tango
```

```
NONE of the devices are in the 'Copied' state.

NONE of the devices are in the 'Copied' state.

NOT ALL of the devices are in the 'Copied' state.

ALL of the devices are in the 'Copied' state.
```

A subsequent `symrcopy query` command indicates that the sessions for the device pairs defined in the file `tango` are now in the Copied state and that copying is 100% complete.

```
symrcopy query -file tango
```

```
Device File Name      : tango

       Control Device                  Remote Device             Flags     Status       Done
---------------------------- ----------------------------------- ----- --------------   ----
                   Protected
```

```
SID:symdev         Tracks     Identification                   RI CDSHUTZ CTL <=> REM     (%)
------------------ ---------  -------------------------------- -- ----- --------------   ----
000187900041:000E9          0 000000003143:0077                SD X..XXS. Copied          100
000187900041:000EA          0 000000003143:0078                SD X..XXS. Copied          100
000187900041:000EB          0 000000003143:0079                SD X..XXS. Copied          100

Total              ---------
  Track(s)                 0
  MB(s)                  0.0


Legend:
         R (Remote Device Vendor Identification)
         S = Symmetrix, . = Unknown.

         I:  (Remote Device Specification Identifier)
         D = Device Name, W = LUN WWN, World Wide Name.

         Flags:
         (C): X = The background copy setting is active for this pair.
              . = The background copy setting is not active for this pair.
         (D): X = The session is a differential copy session.
              . = The session is not a differential copy session.
         (S): X = The session is pushing data to the remote device(s).
              . = The session is pulling data from the remote device(s).
         (H): X = The session is a hot copy session.
              . = The session is a cold copy session.
         (U): X = The session has donor update enabled.
              . = The session does not have donor update enabled.
         (T): M = The session is a migration session.
              R = The session is a RecoverPoint session.
              S = The session is a standard ORS session.
         (Z): X = The session has front-end zero detection enabled.
              . = The session does not have front-end zero detection enabled.
         (*): The failed session can be reactivated.
```

The `symrcopy list` command displays the three inactive copy sessions on the control array whose `sid` is 000187900041 (abbreviated as `41`).

symrcopy list -sid 41

```
Symmetrix ID: 000187900041

Control Device            Remote Device             Flags     Status       Done
--------------- ---------------------------------- ----- --------------   ----
       Protected
Sym    Tracks    Identification                   RI CDSHUTZ CTL <=> REM     (%)
----- ---------  -------------------------------  -- ----- --------------   ----
000E9          0 000000003143:0077                SD X..XXS. Copied          100
000EA          0 000000003143:0078                SD X..XXS. Copied          100
000EB          0 000000003143:0079                SD X..XXS. Copied          100

Total   -------
  Tracks      0
  MB(s)     0.0

Legend:
         R (Remote Device Vendor Identification)
         S = Symmetrix, . = Unknown.

         I:  (Remote Device Specification Identifier)
         D = Device Name, W = LUN WWN, World Wide Name.

         Flags:
         (C): X = The background copy setting is active for this pair.
              . = The background copy setting is not active for this pair.
```

```
(D): X = The session is a differential copy session.
     . = The session is not a differential copy session.
(S): X = The session is pushing data to the remote device(s).
     . = The session is pulling data from the remote device(s).
(H): X = The session is a hot copy session.
     . = The session is a cold copy session.
(U): X = The session has donor update enabled.
     . = The session does not have donor update enabled.
(T): M = The session is a migration session.
     R = The session is a RecoverPoint session.
     S = The session is a standard ORS session.
(Z): X = The session has front-end zero detection enabled.
     . = The session does not have front-end zero detection enabled.
(*): The failed session can be reactivated.
```

Terminating migration session

The `symrcopy terminate` command ends all copy sessions defined in the file `tango`.

`symrcopy terminate -file tango -noprompt`

```
'Terminate' operation execution is in progress for the device list in
device file 'tango'. Please wait...

'Terminate' operation successfully executed for the device list
in device file 'tango'.
```

`A`other `symrcopy list` command verifies that there are no longer any copy sessions on the control array.

`symrcopy list -sid 41`

```
Symmetrix ID: 000187900041
  No Devices with RCopy sessions were found.
```

With the copy operation complete, the remote application on the remote host can be started. However, any changes to remote data at this point are not migrated to the control array unless another full Open Replicator pull operation is performed.

Creating device file

The control devices (`04B3` – `04C2`) and the remote device WWNs are defined in the device text file. The following command uses the `vi` text editor to create a text file named `production01`. The control array `000190101982` is abbreviated as `82`.

```
vi production01

symdev=82:04B3 wwn=60060160F2810D00FC7ED16CC3D5D811
symdev=82:04B4 wwn=60060160F2810D00FD7ED16CC3D5D811
symdev=82:04B5 wwn=60060160F2810D00FE7ED16CC3D5D811
symdev=82:04B6 wwn=60060160F2810D00FF7ED16CC3D5D811
symdev=82:04B7 wwn=60060160F2810D00007FD16CC3D5D811
symdev=82:04B8 wwn=60060160F2810D00017FD16CC3D5D811
symdev=82:04B9 wwn=60060160F2810D00F859A07BC3D5D811
symdev=82:04BA wwn=60060160F2810D00F959A07BC3D5D811
symdev=82:04BB wwn=60060160F2810D007C32B313BE00D911
symdev=82:04BC wwn=60060160F2810D007D32B313BE00D911
symdev=82:04BD wwn=60060160F2810D007E32B313BE00D911
symdev=82:04BE wwn=60060160F2810D007F32B313BE00D911
symdev=82:04BF wwn=60060160F2810D008032B313BE00D911
symdev=82:04C0 wwn=60060160F2810D008132B313BE00D911
symdev=82:04C1 wwn=60060160F2810D007C40DA1ABE00D911
symdev=82:04C2 wwn=60060160F2810D007D40DA1ABE00D911
```

Creating migration session

The `symrcopy create` command is entered from the control host to set up the Open Replicator hot push operation. The command creates 16 online copy sessions, one for each device pair listed in the device file (`production01`). The `-push` parameter specifies that the control array is pushing the data to the remote array. The `-hot` parameter indicates that the array application remains online during the operation. The `-precopy` option indicates that data will immediately begin copying in the background before the session is activated. Background copying in the Precopy state begins as soon as the command line is entered. Refer to "Background copying" on page 609 for information on the Precopy state:

```
symrcopy -file production01 create -precopy -hot -push -noprompt

'Create' operation execution is in progress for the device list
in device file 'production01'. Please wait...

'Create' operation successfully executed for the device list
in device file 'production01'.
```

Querying migration session

The `symrcopy query` command indicates that the sessions for the device pairs defined in the file `production01` are in the Precopy state. To display more detail, include the `-detail` option. To see the full WWN identifier of each remote device, include the `-wwn` option:

```
symrcopy -file production01 query

Device File Name      : production01

       Control Device                  Remote Device             Flags     Status     Done
---------------------------- ----------------------------------- ----- -------------- ----
                    Protected
SID:symdev           Tracks    Identification                     RI  CDSHUTZ CTL <=> REM    (%)
------------------ --------- -------------------------------   --  ----- -------------- ----
000190101982:004B3     275252  60060160F2810D00FC7ED16CC3D5D811 SW X.XX.S. Precopy          30
000190101982:004B4     275256  60060160F2810D00FD7ED16CC3D5D811 SW X.XX.S. Precopy          30
000190101982:004B5     275228  60060160F2810D00FE7ED16CC3D5D811 SW X.XX.S. Precopy          30
000190101982:004B6     275208  60060160F2810D00FF7ED16CC3D5D811 SW X.XX.S. Precopy          30
000190101982:004B7     275196  60060160F2810D00007FD16CC3D5D811 SW X.XX.S. Precopy          30
000190101982:004B8     275220  60060160F2810D00017FD16CC3D5D811 SW X.XX.S. Precopy          30
000190101982:004B9     275234  60060160F2810D00F859A07BC3D5D811 SW X.XX.S. Precopy          30
000190101982:004BA     275284  60060160F2810D00F959A07BC3D5D811 SW X.XX.S. Precopy          30
000190101982:004BB     275194  60060160F2810D007C32B313BE00D911 SW X.XX.S. Precopy          30
000190101982:004BC     275248  60060160F2810D007D32B313BE00D911 SW X.XX.S. Precopy          30
000190101982:004BD     275258  60060160F2810D007E32B313BE00D911 SW X.XX.S. Precopy          30
000190101982:004BE     275206  60060160F2810D007F32B313BE00D911 SW X.XX.S. Precopy          30
000190101982:004BF     275244  60060160F2810D008032B313BE00D911 SW X.XX.S. Precopy          30
000190101982:004C0     275280  60060160F2810D008132B313BE00D911 SW X.XX.S. Precopy          30
000190101982:004C1     275278  60060160F2810D007C40DA1ABE00D911 SW X.XX.S. Precopy          30
000190101982:004C2     275254  60060160F2810D007D40DA1ABE00D911 SW X.XX.S. Precopy          30

Total               ---------
  Track(s)            4403840
  MB(s)                137620

Legend:
         R (Remote Device Vendor Identification)
         S = Symmetrix, . = Unknown.
```

```
I:  (Remote Device Specification Identifier)
D = Device Name, W = LUN WWN, World Wide Name.

Flags:
(C): X = The background copy setting is active for this pair.
     . = The background copy setting is not active for this pair.
(D): X = The session is a differential copy session.
     . = The session is not a differential copy session.
(S): X = The session is pushing data to the remote device(s).
     . = The session is pulling data from the remote device(s).
(H): X = The session is a hot copy session.
     . = The session is a cold copy session.
(U): X = The session has donor update enabled.
     . = The session does not have donor update enabled.
(T): M = The session is a migration session.
     R = The session is a RecoverPoint session.
     S = The session is a standard ORS session.
(Z): X = The session has front-end zero detection enabled.
     . = The session does not have front-end zero detection enabled.
(*): The failed session can be reactivated.
```

Activating migration session

When the session is activated using the `symrcopy activate` command, the copy operation continues in the CopyInProg state. Using the `-consistent` option creates a consistent point-in-time copy:

```
symrcopy -file production01 activate -consistent -noprompt

'Activate' operation execution is in progress for the device list
in device file 'production01'. Please wait...

'Activate' operation successfully executed for the device list
in device file 'production01'.
```

Entering the `symrcopy query` command again indicates that the sessions for the device pairs defined in the file `production01` are in the CopyInProg state:

```
symrcopy -file production01 query

Device File Name      : production01

       Control Device                  Remote Device             Flags     Status     Done
---------------------------- ---------------------------------- ----- -------------- ----
                   Protected
SID:symdev         Tracks    Identification                   RI  CDSHUTZ CTL <=> REM     (%)
------------------ --------- -------------------------------- --  ----- -------------- ----
000190101982:004B3     267170 60060160F2810D00FC7ED16CC3D5D811 SW  X.XX.S. CopyInProg       60
000190101982:004B4     267220 60060160F2810D00FD7ED16CC3D5D811 SW  X.XX.S. CopyInProg       60
000190101982:004B5     267146 60060160F2810D00FE7ED16CC3D5D811 SW  X.XX.S. CopyInProg       60
000190101982:004B6     267188 60060160F2810D00FF7ED16CC3D5D811 SW  X.XX.S. CopyInProg       60
000190101982:004B7     267178 60060160F2810D00007FD16CC3D5D811 SW  X.XX.S. CopyInProg       60
000190101982:004B8     267082 60060160F2810D00017FD16CC3D5D811 SW  X.XX.S. CopyInProg       60
000190101982:004B9     267218 60060160F2810D00F859A07BC3D5D811 SW  X.XX.S. CopyInProg       60
000190101982:004BA     267226 60060160F2810D00F959A07BC3D5D811 SW  X.XX.S. CopyInProg       60
000190101982:004BB     267068 60060160F2810D007C32B313BE00D911 SW  X.XX.S. CopyInProg       60
000190101982:004BC     267128 60060160F2810D007D32B313BE00D911 SW  X.XX.S. CopyInProg       60
000190101982:004BD     267106 60060160F2810D007E32B313BE00D911 SW  X.XX.S. CopyInProg       60
000190101982:004BE     267214 60060160F2810D007F32B313BE00D911 SW  X.XX.S. CopyInProg       60
000190101982:004BF     267240 60060160F2810D008032B313BE00D911 SW  X.XX.S. CopyInProg       60
000190101982:004C0     267184 60060160F2810D008132B313BE00D911 SW  X.XX.S. CopyInProg       60
000190101982:004C1     267100 60060160F2810D007C40DA1ABE00D911 SW  X.XX.S. CopyInProg       60
000190101982:004C2     267236 60060160F2810D007D40DA1ABE00D911 SW  X.XX.S. CopyInProg       60
```

```
Total              ---------
  Track(s)           4274704
  MB(s)               133584

Legend:
         R (Remote Device Vendor Identification)
         S = Symmetrix, . = Unknown.

         I:  (Remote Device Specification Identifier)
         D = Device Name, W = LUN WWN, World Wide Name.

         Flags:
         (C): X = The background copy setting is active for this pair.
              . = The background copy setting is not active for this pair.
         (D): X = The session is a differential copy session.
              . = The session is not a differential copy session.
         (S): X = The session is pushing data to the remote device(s).
              . = The session is pulling data from the remote device(s).
         (H): X = The session is a hot copy session.
              . = The session is a cold copy session.
         (U): X = The session has donor update enabled.
              . = The session does not have donor update enabled.
         (T): M = The session is a migration session.
              R = The session is a RecoverPoint session.
              S = The session is a standard ORS session.
         (Z): X = The session has front-end zero detection enabled.
              . = The session does not have front-end zero detection enabled.
         (*): The failed session can be reactivated.
```

A later entry of the `symrcopy query` command indicates that the sessions for the device pairs defined in the file `production01` have completed data copying and are in the Copied state:

symrcopy -file production01 query

```
Device File Name      : production01

       Control Device                  Remote Device             Flags     Status        Done
---------------------------- ----------------------------------- ----- --------------    ----
                   Protected
SID:symdev         Tracks     Identification                   RI  CDSHUTZ CTL <=> REM     (%)
------------------ --------- -------------------------------- --  -----   -------------- ----
000190101982:004B3          0 60060160F2810D00FC7ED16CC3D5D811 SW  X.XX.S. Copied          100
000190101982:004B4          0 60060160F2810D00FD7ED16CC3D5D811 SW  X.XX.S. Copied          100
000190101982:004B5          0 60060160F2810D00FE7ED16CC3D5D811 SW  X.XX.S. Copied          100
000190101982:004B6          0 60060160F2810D00FF7ED16CC3D5D811 SW  X.XX.S. Copied          100
000190101982:004B7          0 60060160F2810D00007FD16CC3D5D811 SW  X.XX.S. Copied          100
000190101982:004B8          0 60060160F2810D00017FD16CC3D5D811 SW  X.XX.S. Copied          100
000190101982:004B9          0 60060160F2810D00F859A07BC3D5D811 SW  X.XX.S. Copied          100
000190101982:004BA          0 60060160F2810D00F959A07BC3D5D811 SW  X.XX.S. Copied          100
000190101982:004BB          0 60060160F2810D007C32B313BE00D911 SW  X.XX.S. Copied          100
000190101982:004BC          0 60060160F2810D007D32B313BE00D911 SW  X.XX.S. Copied          100
000190101982:004BD          0 60060160F2810D007E32B313BE00D911 SW  X.XX.S. Copied          100
000190101982:004BE          0 60060160F2810D007F32B313BE00D911 SW  X.XX.S. Copied          100
000190101982:004BF          0 60060160F2810D008032B313BE00D911 SW  X.XX.S. Copied          100
000190101982:004C0          0 60060160F2810D008132B313BE00D911 SW  X.XX.S. Copied          100
000190101982:004C1          0 60060160F2810D007C40DA1ABE00D911 SW  X.XX.S. Copied          100
000190101982:004C2          0 60060160F2810D007D40DA1ABE00D911 SW  X.XX.S. Copied          100

Total              ---------
  Track(s)                 0
  MB(s)                  0.0

Legend:
         R (Remote Device Vendor Identification)
```

```
S = Symmetrix, . = Unknown.

I:  (Remote Device Specification Identifier)
D = Device Name, W = LUN WWN, World Wide Name.

Flags:
(C): X = The background copy setting is active for this pair.
     . = The background copy setting is not active for this pair.
(D): X = The session is a differential copy session.
     . = The session is not a differential copy session.
(S): X = The session is pushing data to the remote device(s).
     . = The session is pulling data from the remote device(s).
(H): X = The session is a hot copy session.
     . = The session is a cold copy session.
(U): X = The session has donor update enabled.
     . = The session does not have donor update enabled.
(T): M = The session is a migration session.
     R = The session is a RecoverPoint session.
     S = The session is a standard ORS session.
(Z): X = The session has front-end zero detection enabled.
     . = The session does not have front-end zero detection enabled.
(*): The failed session can be reactivated.
```

Creating migration session with differential copying

The following example shows the same sessions created again. Sessions are then recreated and restored. In order for sessions to be recreated or restored, they must originally be created with differential copying. This functionality is available only for push operations.

Note: Hot push sessions are created with `-differential` option as the default, so this option does not need to be specified with the `create` command.

The `symrcopy create` command is entered from the control host to set up the Open Replicator hot push operation. The session will perform incremental copies, capturing only new writes to the control devices. The `-precopy` option indicates that data will immediately begin copying in the background before the session is activated:

```
symrcopy -file production01 create -precopy -hot -push -noprompt

'Create' operation execution is in progress for the device list
in device file 'production01'. Please wait...

'Create' operation successfully executed for the device list
in device file 'production01'.
```

The session is then activated using the `symrcopy activate` command with the `-consistent` option to create a consistent point-in-time copy:

```
symrcopy -file production01 activate -consistence -noprompt

'Activate' operation execution is in progress for the device list
in device file 'production01'. Please wait...

'Activate' operation successfully executed for the device list
in device file 'production01'.
```

A subsequent `symrcopy query` command shows that the sessions for the device pairs defined in the file `production01` are in the CopyInProg state. The Flags indicate that the sessions are differential copy sessions:

```
symrcopy -file production01 query

Device File Name      : production01

       Control Device                  Remote Device             Flags     Status     Done
---------------------------- ---------------------------------- ----- -------------- ----
                   Protected
SID:symdev         Tracks     Identification                  RI  CDSHUTZ CTL <=> REM     (%)
------------------ --------- -------------------------------- --  ----- -------------- ----
000190101982:004B3     273226 60060160F2810D00FC7ED16CC3D5D811 SW  XXXX.S. CopyInProg       30
000190101982:004B4     273148 60060160F2810D00FD7ED16CC3D5D811 SW  XXXX.S. CopyInProg       30
000190101982:004B5     273236 60060160F2810D00FE7ED16CC3D5D811 SW  XXXX.S. CopyInProg       30
000190101982:004B6     273242 60060160F2810D00FF7ED16CC3D5D811 SW  XXXX.S. CopyInProg       30
000190101982:004B7     273236 60060160F2810D00007FD16CC3D5D811 SW  XXXX.S. CopyInProg       30
000190101982:004B8     273246 60060160F2810D00017FD16CC3D5D811 SW  XXXX.S. CopyInProg       30
000190101982:004B9     273214 60060160F2810D00F859A07BC3D5D811 SW  XXXX.S. CopyInProg       30
000190101982:004BA     273206 60060160F2810D00F959A07BC3D5D811 SW  XXXX.S. CopyInProg       30
000190101982:004BB     273268 60060160F2810D007C32B313BE00D911 SW  XXXX.S. CopyInProg       30
000190101982:004BC     273248 60060160F2810D007D32B313BE00D911 SW  XXXX.S. CopyInProg       30
000190101982:004BD     273260 60060160F2810D007E32B313BE00D911 SW  XXXX.S. CopyInProg       30
000190101982:004BE     273224 60060160F2810D007F32B313BE00D911 SW  XXXX.S. CopyInProg       30
000190101982:004BF     273162 60060160F2810D008032B313BE00D911 SW  XXXX.S. CopyInProg       30
000190101982:004C0     273286 60060160F2810D008132B313BE00D911 SW  XXXX.S. CopyInProg       30
000190101982:004C1     273182 60060160F2810D007C40DA1ABE00D911 SW  XXXX.S. CopyInProg       30
000190101982:004C2     273196 60060160F2810D007D40DA1ABE00D911 SW  XXXX.S. CopyInProg       30

Total              ---------
  Track(s)           4371580
  MB(s)               136612

Legend:
        R (Remote Device Vendor Identification)
        S = Symmetrix, . = Unknown.

        I:  (Remote Device Specification Identifier)
        D = Device Name, W = LUN WWN, World Wide Name.

        Flags:
        (C): X = The background copy setting is active for this pair.
             . = The background copy setting is not active for this pair.
        (D): X = The session is a differential copy session.
             . = The session is not a differential copy session.
        (S): X = The session is pushing data to the remote device(s).
             . = The session is pulling data from the remote device(s).
        (H): X = The session is a hot copy session.
             . = The session is a cold copy session.
        (U): X = The session has donor update enabled.
             . = The session does not have donor update enabled.
        (T): M = The session is a migration session.
             R = The session is a RecoverPoint session.
             S = The session is a standard ORS session.
        (Z): X = The session has front-end zero detection enabled.
             . = The session does not have front-end zero detection enabled.
        (*): The failed session can be reactivated.
```

A subsequent query operation shows that the sessions for the device pairs defined in the file `production01` have completed copying data and are in the Copied state:

```
symrcopy -file production01 query

Device File Name      : production01

       Control Device                  Remote Device             Flags     Status     Done
---------------------------- ---------------------------------- ----- -------------- ----
                   Protected
SID:symdev         Tracks     Identification                  RI  CDSHUTZ CTL <=> REM (%)
```

```
------------------ ---------  -------------------------------- --  ----- -------------- ----
000190101982:004B3          0 60060160F2810D00FC7ED16CC3D5D811 SW  XXXX.S. Copied 100
000190101982:004B4          0 60060160F2810D00FD7ED16CC3D5D811 SW  XXXX.S. Copied 100
000190101982:004B5          0 60060160F2810D00FE7ED16CC3D5D811 SW  XXXX.S. Copied 100
000190101982:004B6          0 60060160F2810D00FF7ED16CC3D5D811 SW  XXXX.S. Copied 100
000190101982:004B7          0 60060160F2810D00007FD16CC3D5D811 SW  XXXX.S. Copied 100
000190101982:004B8          0 60060160F2810D00017FD16CC3D5D811 SW  XXXX.S. Copied 100
000190101982:004B9          0 60060160F2810D00F859A07BC3D5D811 SW  XXXX.S. Copied 100
000190101982:004BA          0 60060160F2810D00F959A07BC3D5D811 SW  XXXX.S. Copied 100
000190101982:004BB          0 60060160F2810D007C32B313BE00D911 SW  XXXX.S. Copied 100
000190101982:004BC          0 60060160F2810D007D32B313BE00D911 SW  XXXX.S. Copied 100
000190101982:004BD          0 60060160F2810D007E32B313BE00D911 SW  XXXX.S. Copied 100
000190101982:004BE          0 60060160F2810D007F32B313BE00D911 SW  XXXX.S. Copied 100
000190101982:004BF          0 60060160F2810D008032B313BE00D911 SW  XXXX.S. Copied 100
000190101982:004C0          0 60060160F2810D008132B313BE00D911 SW  XXXX.S. Copied 100
000190101982:004C1          0 60060160F2810D007C40DA1ABE00D911 SW  XXXX.S. Copied 100
000190101982:004C2          0 60060160F2810D007D40DA1ABE00D911 SW  XXXX.S. Copied 100

Total              ---------
  Track(s)                 0
  MB(s)                  0.0

Legend:
         R (Remote Device Vendor Identification)
         S = Symmetrix, . = Unknown.

         I:  (Remote Device Specification Identifier)
         D = Device Name, W = LUN WWN, World Wide Name.

         Flags:
         (C): X = The background copy setting is active for this pair.
              . = The background copy setting is not active for this pair.
         (D): X = The session is a differential copy session.
              . = The session is not a differential copy session.
         (S): X = The session is pushing data to the remote device(s).
              . = The session is pulling data from the remote device(s).
         (H): X = The session is a hot copy session.
              . = The session is a cold copy session.
         (U): X = The session has donor update enabled.
              . = The session does not have donor update enabled.
         (T): M = The session is a migration session.
              R = The session is a RecoverPoint session.
              S = The session is a standard ORS session.
         (Z): X = The session has front-end zero detection enabled.
              . = The session does not have front-end zero detection enabled.
         (*): The failed session can be reactivated.
```

Recreating migration session for incremental update

At a later time, it is desired to recreate the session to incrementally update the remote devices to copy any device tracks that were changed since the time that the session actively finished copying. The `symrcopy recreate` command recreates the `production01` copy session for the control devices and sets it up for a differential copy operation:

```
symrcopy -file production01 recreate -noprompt

'Recreate' operation execution is in progress for the device list
in device file 'production01'. Please wait...

'Recreate' operation successfully executed for the device list
in device file 'production01'.
```

The `symrcopy query` command shows that the control/remote pairs are in the Recreated state:

symrcopy -file production01 query

```
Device File Name      : production01

       Control Device                  Remote Device             Flags     Status     Done
---------------------------- ----------------------------------- ----- -------------- ----
                      Protected
SID:symdev            Tracks    Identification                   RI  CDSHUTZ CTL <=> REM     (%)
------------------  --------- -------------------------------- --  ----- -------------- ----
000190101982:004B3          0 60060160F2810D00FC7ED16CC3D5D811 SW  XXXX.S. Recreated       N/A
000190101982:004B4          0 60060160F2810D00FD7ED16CC3D5D811 SW  XXXX.S. Recreated       N/A
000190101982:004B5          0 60060160F2810D00FE7ED16CC3D5D811 SW  XXXX.S. Recreated       N/A
000190101982:004B6          0 60060160F2810D00FF7ED16CC3D5D811 SW  XXXX.S. Recreated       N/A
000190101982:004B7          0 60060160F2810D00007FD16CC3D5D811 SW  XXXX.S. Recreated       N/A
000190101982:004B8          0 60060160F2810D00017FD16CC3D5D811 SW  XXXX.S. Recreated       N/A
000190101982:004B9          0 60060160F2810D00F859A07BC3D5D811 SW  XXXX.S. Recreated       N/A
000190101982:004BA          0 60060160F2810D00F959A07BC3D5D811 SW  XXXX.S. Recreated       N/A
000190101982:004BB          0 60060160F2810D007C32B313BE00D911 SW  XXXX.S. Recreated       N/A
000190101982:004BC          0 60060160F2810D007D32B313BE00D911 SW  XXXX.S. Recreated       N/A
000190101982:004BD          0 60060160F2810D007E32B313BE00D911 SW  XXXX.S. Recreated       N/A
000190101982:004BE          0 60060160F2810D007F32B313BE00D911 SW  XXXX.S. Recreated       N/A
000190101982:004BF          0 60060160F2810D008032B313BE00D911 SW  XXXX.S. Recreated       N/A
000190101982:004C0          0 60060160F2810D008132B313BE00D911 SW  XXXX.S. Recreated       N/A
000190101982:004C1          0 60060160F2810D007C40DA1ABE00D911 SW  XXXX.S. Recreated       N/A
000190101982:004C2          0 60060160F2810D007D40DA1ABE00D911 SW  XXXX.S. Recreated       N/A

Total               ---------
  Track(s)                  0
  MB(s)                   0.0

Legend:
         R (Remote Device Vendor Identification)
         S = Symmetrix, . = Unknown.

         I:  (Remote Device Specification Identifier)
         D = Device Name, W = LUN WWN, World Wide Name.

         Flags:
         (C): X = The background copy setting is active for this pair.
              . = The background copy setting is not active for this pair.
         (D): X = The session is a differential copy session.
              . = The session is not a differential copy session.
         (S): X = The session is pushing data to the remote device(s).
              . = The session is pulling data from the remote device(s).
         (H): X = The session is a hot copy session.
              . = The session is a cold copy session.
         (U): X = The session has donor update enabled.
              . = The session does not have donor update enabled.
         (T): M = The session is a migration session.
              R = The session is a RecoverPoint session.
              S = The session is a standard ORS session.
         (Z): X = The session has front-end zero detection enabled.
              . = The session does not have front-end zero detection enabled.
         (*): The failed session can be reactivated.
```

Once the recreated session is activated, the incremental update of devices begins, copying any device tracks that were changed since the last time the copy session actively finished copying:

symrcopy -file production01 activate -noprompt

```
'Activate' operation execution is in progress for the device list
```

```
in device file 'production01'. Please wait...

'Activate' operation successfully executed for the device list
in device file 'production01'.
```

A later query operation shows that the devices have finished copying the incremental data:

symrcopy -file production01 query

```
Device File Name      : production01

       Control Device                  Remote Device             Flags     Status     Done
---------------------------- ----------------------------------- ----- -------------- ----
                    Protected
SID:symdev          Tracks     Identification                   RI  CDSHUTZ CTL <=> REM (%)
------------------ --------- -------------------------------- --  ----- -------------- ----
000190101982:004B3          0 60060160F2810D00FC7ED16CC3D5D811 SW  XXXX.S. Copied       100
000190101982:004B4          0 60060160F2810D00FD7ED16CC3D5D811 SW  XXXX.S. Copied       100
000190101982:004B5          0 60060160F2810D00FE7ED16CC3D5D811 SW  XXXX.S. Copied       100
000190101982:004B6          0 60060160F2810D00FF7ED16CC3D5D811 SW  XXXX.S. Copied       100
000190101982:004B7          0 60060160F2810D00007FD16CC3D5D811 SW  XXXX.S. Copied       100
000190101982:004B8          0 60060160F2810D00017FD16CC3D5D811 SW  XXXX.S. Copied       100
000190101982:004B9          0 60060160F2810D00F859A07BC3D5D811 SW  XXXX.S. Copied       100
000190101982:004BA          0 60060160F2810D00F959A07BC3D5D811 SW  XXXX.S. Copied       100
000190101982:004BB          0 60060160F2810D007C32B313BE00D911 SW  XXXX.S. Copied       100
000190101982:004BC          0 60060160F2810D007D32B313BE00D911 SW  XXXX.S. Copied       100
000190101982:004BD          0 60060160F2810D007E32B313BE00D911 SW  XXXX.S. Copied       100
000190101982:004BE          0 60060160F2810D007F32B313BE00D911 SW  XXXX.S. Copied       100
000190101982:004BF          0 60060160F2810D008032B313BE00D911 SW  XXXX.S. Copied       100
000190101982:004C0          0 60060160F2810D008132B313BE00D911 SW  XXXX.S. Copied       100
000190101982:004C1          0 60060160F2810D007C40DA1ABE00D911 SW  XXXX.S. Copied       100
000190101982:004C2          0 60060160F2810D007D40DA1ABE00D911 SW  XXXX.S. Copied       100

Total               ---------
  Track(s)                  0
  MB(s)                   0.0

Legend:
         R (Remote Device Vendor Identification)
         S = Symmetrix, . = Unknown.

         I:  (Remote Device Specification Identifier)
         D = Device Name, W = LUN WWN, World Wide Name.

         Flags:
         (C): X = The background copy setting is active for this pair.
              . = The background copy setting is not active for this pair.
         (D): X = The session is a differential copy session.
              . = The session is not a differential copy session.
         (S): X = The session is pushing data to the remote device(s).
              . = The session is pulling data from the remote device(s).
         (H): X = The session is a hot copy session.
              . = The session is a cold copy session.
         (U): X = The session has donor update enabled.
              . = The session does not have donor update enabled.
         (T): M = The session is a migration session.
              R = The session is a RecoverPoint session.
              S = The session is a standard ORS session.
         (Z): X = The session has front-end zero detection enabled.
              . = The session does not have front-end zero detection enabled.
         (*): The failed session can be reactivated.
```

In the scenario described above, differential copy sessions were copied completely and then recreated to capture any additional changes made to the control devices since the last copy. This has created a point-in-time copy on the remote devices.

Restoring data to control device

At a later time, it was discovered that a possible data corruption had occurred, which may have affected data on the control devices. To correct the issue, it has been decided that the last incremental copy of data to the remote devices should be restored back to the control devices.

To recover the original data from the remote devices since the last activation of the sessions, use the `symrcopy restore` command. For additional information, refer to "Restoring a session" on page 619:

```
symrcopy -file production01 restore -noprompt

'Incr Restore' operation execution is in progress for the device list
in device file 'production01'. Please wait...

'Incr Restore' operation successfully executed for the device list
in device file 'production01'.
```

A subsequent query operation shows that the control/remote pairs are now in the Restored state. Data has been 100% restored back to the control devices from the last time the sessions were activated for the differential recreate. Note that the flags identify that data is being pulled from the remote devices back to the control devices (S):

```
symrcopy -file production01 query


Device File Name      : production01

       Control Device                  Remote Device             Flags     Status     Done
---------------------------- ----------------------------------- ----- -------------- ----
                    Protected
SID:symdev          Tracks    Identification                   RI  CDSHUTZ CTL <=> REM     (%)
------------------- --------- ------------------------------- --  ----- -------------- ----
000190101982:004B3          0 60060160F2810D00FC7ED16CC3D5D811 SW  XX.X.S. Restored        100
000190101982:004B4          0 60060160F2810D00FD7ED16CC3D5D811 SW  XX.X.S. Restored        100
000190101982:004B5          0 60060160F2810D00FE7ED16CC3D5D811 SW  XX.X.S. Restored        100
000190101982:004B6          0 60060160F2810D00FF7ED16CC3D5D811 SW  XX.X.S. Restored        100
000190101982:004B7          0 60060160F2810D00007FD16CC3D5D811 SW  XX.X.S. Restored        100
000190101982:004B8          0 60060160F2810D00017FD16CC3D5D811 SW  XX.X.S. Restored        100
000190101982:004B9          0 60060160F2810D00F859A07BC3D5D811 SW  XX.X.S. Restored        100
000190101982:004BA          0 60060160F2810D00F959A07BC3D5D811 SW  XX.X.S. Restored        100
000190101982:004BB          0 60060160F2810D007C32B313BE00D911 SW  XX.X.S. Restored        100
000190101982:004BC          0 60060160F2810D007D32B313BE00D911 SW  XX.X.S. Restored        100
000190101982:004BD          0 60060160F2810D007E32B313BE00D911 SW  XX.X.S. Restored        100
000190101982:004BE          0 60060160F2810D007F32B313BE00D911 SW  XX.X.S. Restored        100
000190101982:004BF          0 60060160F2810D008032B313BE00D911 SW  XX.X.S. Restored        100
000190101982:004C0          0 60060160F2810D008132B313BE00D911 SW  XX.X.S. Restored        100
000190101982:004C1          0 60060160F2810D007C40DA1ABE00D911 SW  XX.X.S. Restored        100
000190101982:004C2          0 60060160F2810D007D40DA1ABE00D911 SW  XX.X.S. Restored        100

Total               ---------
  Track(s)                  0
  MB(s)                   0.0

Legend:
         R (Remote Device Vendor Identification)
         S = Symmetrix, . = Unknown.
```

```
I:  (Remote Device Specification Identifier)
D = Device Name, W = LUN WWN, World Wide Name.

Flags:
(C): X = The background copy setting is active for this pair.
     . = The background copy setting is not active for this pair.
(D): X = The session is a differential copy session.
     . = The session is not a differential copy session.
(S): X = The session is pushing data to the remote device(s).
     . = The session is pulling data from the remote device(s).
(H): X = The session is a hot copy session.
     . = The session is a cold copy session.
(U): X = The session has donor update enabled.
     . = The session does not have donor update enabled.
(T): M = The session is a migration session.
     R = The session is a RecoverPoint session.
     S = The session is a standard ORS session.
(Z): X = The session has front-end zero detection enabled.
     . = The session does not have front-end zero detection enabled.
(*): The failed session can be reactivated.
```

The `symrcopy terminate` command ends all copy sessions defined in the file `production01`:

```
symrcopy terminate -file production01 -noprompt

'Terminate' operation execution is in progress for the device
list in device file 'production01'. Please wait...

'Terminate' operation successfully executed for the device list
in the device file 'production01'.
```

Example 2: Pushing data using BCVs as the control devices

This example shows how to push data from a control array using TimeFinder BCVs as the control devices. The example also shows how to initially get the full data to the remote devices using the BCVs, then repeats it incrementally.

The following examples and tasks show:

1. “Creating device group”
2. “Creating device file”
3. “Establishing TimeFinder session”
4. “Splitting TimeFinder session”
5. “Creating migration session”
6. “Activating migration session”
7. “Querying migration session”
8. “Checking for invalid tracks on standard device”
9. “Re-establishing TimeFinder session”
10. “Recreating migration session for incremental update”
11. “Verifying session status”

Note: For arrays running HYPERMAX OS 5977, push operations are not supported.

Prerequisite tasks

An important setup requirement is that the BCVs be mapped to an FA, and that FA must be zoned to the remote storage array. Also, the FA should not be the same FA for the control standard devices. The example uses two standard devices and two BCV devices on a control array (`sid 35`) running Enginuity version 5671, and two remote devices on a remote array (`sid 60`).

Creating device group

To perform TimeFinder operations, the example first needs to create a device group (`conga`), add the two standard devices (`86` and `87`), and associate the BCV devices (`B8` and `B9`) to the device group:

```
symdg create conga
symdg -g conga addall dev -range 86:87 -sid 35
symbcv -g conga associateall dev -range B8:B9 -sid 35
```

Creating device file

The `vi` command uses the vi text editor to create a text file (also named `conga`) to set up the Open Replicator copy pairings. For example, the first pairing is BCV B8 on array 35 with a remote device whose LUN WWN is 6006048000018450016053594D374646.

```
vi conga

symdev=35:B8  wwn=6006048000018450016053594D374646
symdev=35:B9  wwn=6006048000018450016053594D383030
```

Establishing TimeFinder session

The TimeFinder `symclone establish` command synchronizes the control array's BCV pairs defined in device group `conga`. The result is a full copy of the data from the standard devices to its BCV devices:

```
symclone -g conga establish -full -noprompt

'Full Establish' operation execution is in progress for
device group 'conga'. Please wait...

'Full Establish' operation successfully initiated for device group
   'conga'.
```

The `symclone verify` command checks at 60-second intervals to verify when the BCV pairs are synchronized:

```
symclone -g conga verify -i 60

None of the devices in group 'conga' are in the 'Synchronized or
Restored' state.

None of the devices in group 'conga' are in the 'Synchronized or
Restored' state.

All of the devices in group 'conga' are in the 'Synchronized or
Restored' state.
```

Splitting TimeFinder session

The `symclone split` command with the `-consistent` option splits the control array's BCV pairs so that the data is consistent after the split. To make the BCV inaccessible to the host after the split, use the `-not_ready` option:

```
symclone -g conga split -consistent -noprompt -not_ready

'Split' operation execution is in progress for
device group 'conga'. Please wait...

'Split' operation successfully executed for device group 'conga'.
```

The `symclone verify -split` command checks at 60-second intervals to determine when the BCV pairs have completed splitting in the background (`-bg`):

```
symclone -g conga verify -split -bg -i 60

All of the devices in group 'conga' have finished splitting in the
background.
```

The `symclone query` command shows that the BCVs in the device group are split, which means they are ready to be the control devices in the Open Replicator copy sessions:

```
symclone -g conga  query

Device Group (DG) Name: conga
DG's Type             : REGULAR
DG's Symmetrix ID     : 000187900035


     Standard Device                    BCV Device                  State
------------------------- ------------------------------------- ------------
                    Inv.                                Inv.
Logical        Sym  Tracks Logical              Sym       Tracks STD <=> BCV
------------------------- ------------------------------------- ------------

DEV001         00086      0 BCV001               000B8 *         0 Split
DEV002         00087      0 BCV002               000B9 *         0 Split

Total              -------                               -------
  Track(s)               0                                     0
  MB(s)                0.0                                   0.0


Legend:

(*): The paired BCV device is associated with this group.
```

Creating migration session

The `symrcopy create` command with the `-cold` and `-push` options creates two offline copy sessions for those pairings defined in the `conga` text file. A full and immediate copy is made. All subsequent copying during that copy session automatically performs incremental copies, capturing only new writes to the control device.

It is imperative, however, that the remote devices not be accessed by their host if a subsequent differential push (`recreate` and `activate`) will be performed. If the remote data needs to be accessed during this time, a copy of the data needs to be made on the remote storage array:

```
symrcopy create -file conga -cold -push -noprompt
```

```
'Create' operation execution is in progress for the device list in
device file 'conga'. Please wait...

'Create' operation successfully executed for the device list
in device file 'conga'.
```

Activating migration session

The `symrcopy activate` command starts the copy operation for the device pairs defined in the text file `conga`. Copying from the control devices to the two remote devices begins:

```
symrcopy activate -file conga -noprompt

'Activate' operation execution is in progress for the device list in
  device file 'conga'. Please wait...

'Activate' operation successfully executed for the device list in
  device file 'conga'.
```

The `symrcopy verify` command checks at 60-second intervals to verify whether the control/remote device pairs are in the Copied state:

```
symrcopy verify -file conga -i 60

NONE of the devices are in the 'Copied' state.

NONE of the devices are in the 'Copied' state.

NOT ALL of the devices are in the 'Copied' state.

ALL of the devices are in the 'Copied' state.
```

Querying migration session

The `symrcopy query` command shows that the control/remote device pairs are in the Copied state:

```
symrcopy -file conga query

Device File Name      : conga

       Control Device                  Remote Device             Flags     Status     Done
---------------------------- ---------------------------------- ----- -------------- ----
                   Protected
SID:symdev         Tracks     Identification                   RI CDSHUTZ CTL <=> REM     (%)
------------------ --------- -------------------------------- -- ----- -------------- ----
000187900035:000B8          0 000184500160:07FF                SD XXX..S. Copied          100
000187900035:000B9          0 000184500160:0800                SD XXX..S. Copied          100

Total              ---------
  Track(s)                 0
  MB(s)                  0.0

Legend:
         R (Remote Device Vendor Identification)
         S = Symmetrix, . = Unknown.

         I:  (Remote Device Specification Identifier)
         D = Device Name, W = LUN WWN, World Wide Name.

         Flags:
         (C): X = The background copy setting is active for this pair.
              . = The background copy setting is not active for this pair.
         (D): X = The session is a differential copy session.
              . = The session is not a differential copy session.
```

```
(S): X = The session is pushing data to the remote device(s).
     . = The session is pulling data from the remote device(s).
(H): X = The session is a hot copy session.
     . = The session is a cold copy session.
(U): X = The session has donor update enabled.
     . = The session does not have donor update enabled.
(T): M = The session is a migration session.
     R = The session is a RecoverPoint session.
     S = The session is a standard ORS session.
(Z): X = The session has front-end zero detection enabled.
     . = The session does not have front-end zero detection enabled.
(*): The failed session can be reactivated.
```

Checking for invalid tracks on standard device

A TimeFinder `symmir query` command displays the device group `conga` and shows that changes have occurred (invalid tracks) on the control array's standard devices while they were split from the BCVs:

symmir -g conga query

```
Device Group (DG) Name: conga
DG's Type             : REGULAR
DG's Symmetrix ID     : 000187900035


     Standard Device                    BCV Device                  State
------------------------- ------------------------------------- ------------
                     Inv.                                 Inv.
Logical        Sym  Tracks Logical              Sym       Tracks STD <=> BCV
------------------------- ------------------------------------- ------------

DEV001         00086   3125 BCV001               000B8 *        30 Split
DEV002         00087   3125 BCV002               000B9 *        30 Split

Total              -------                               -------
  Track(s)            6250                                    60
  MB(s)              195.3                                   1.9


Legend:

(*): The paired BCV device is associated with this group.
```

Re-establishing TimeFinder session

The BCVs can be updated with the invalid tracks by using `symmir establish` command to incrementally re-establish the TimeFinder BCV pairs:

symmir -g conga establish -noprompt

```
'Incremental Establish' operation execution is in progress for device
   group 'conga'. Please wait...

'Incremental Establish' operation successfully initiated for device
   group 'conga'.
```

The `symmir verify` command checks at 60-second intervals to verify when the BCV pairs reach the Synchronized state:

symmir -g conga verify -i 60

```
None of the devices in group 'conga' are in the 'Synchronized or
   Restored'
state.
```

```
All of the devices in group 'conga' are in the 'Synchronized or
   Restored' state.
```

The `symmir split` command splits the BCV pairs again in a consistent fashion, making the BCV devices Not Ready to their host:

```
symmir -g conga split -consistent -noprompt -not_ready
```

```
'Split' operation execution is in progress for
device group 'conga'. Please wait...

'Split' operation successfully executed for device group 'conga'.
```

The `symmir verify -split` command verifies again when the BCV pairs are finished splitting in the background:

```
symmir -g conga verify -split -bg
```

```
All of the devices in group 'conga' have finished splitting in the
background.
```

Recreating migration session for incremental update

The `symrcopy recreate` command recreates the copy sessions defined in the text file `conga`:

```
symrcopy recreate -file conga -noprompt
```

```
'Recreate' operation execution is in progress for the device list in
   device file 'conga'. Please wait...

'Recreate' operation successfully executed for the device list in
   device file 'conga'.
```

The `symrcopy query` command shows the control/remote pairs in the Recreated state. Note that the protected track count for the control devices is the same as their invalid track count displayed in the previous `symmir query` command:

```
symrcopy query -file conga
```

```
Device File Name      : conga

       Control Device                  Remote Device             Flags     Status     Done
----------------------------- ----------------------------------- ----- -------------- ----
                    Protected
SID:symdev          Tracks     Identification                   RI CDSHUTZ CTL <=> REM     (%)
------------------  ---------- ---------------------------------- -- ----- -------------- ----
000187900035:000B8       3125 000184500160:07FF                 SD XXX..S. Recreated       N/A
000187900035:000B9       3125 000184500160:0800                 SD XXX..S. Recreated       N/A

Total               ---------
  Track(s)               6250
  MB(s)                 195.3

Legend:
         R (Remote Device Vendor Identification)
         S = Symmetrix, . = Unknown.

         I:  (Remote Device Specification Identifier)
         D = Device Name, W = LUN WWN, World Wide Name.

         Flags:
         (C): X = The background copy setting is active for this pair.
              . = The background copy setting is not active for this pair.
```

```
(D): X = The session is a differential copy session.
     . = The session is not a differential copy session.
(S): X = The session is pushing data to the remote device(s).
     . = The session is pulling data from the remote device(s).
(H): X = The session is a hot copy session.
     . = The session is a cold copy session.
(U): X = The session has donor update enabled.
     . = The session does not have donor update enabled.
(T): M = The session is a migration session.
     R = The session is a RecoverPoint session.
     S = The session is a standard ORS session.
(Z): X = The session has front-end zero detection enabled.
     . = The session does not have front-end zero detection enabled.
(*): The failed session can be reactivated.
```

The `symrcopy activate` command activates the copy sessions. This initiates an incremental copy operation that copies only those device tracks that have changed since the initial full copy was performed:

```
symrcopy activate -file conga -noprompt

'Activate' operation execution is in progress for the device list in
   device file 'conga'. Please wait...

'Activate' operation successfully executed for the device list in
   device file 'conga'.
```

Verifying session status

The `symrcopy verify` command checks at 60-second intervals to verify whether the control/remote device pairs are in the Copied state:

```
symrcopy verify -file conga -i 60

NONE of the devices are in the 'Copied' state.

NONE of the devices are in the 'Copied' state.

NOT ALL of the devices are in the 'Copied' state.

ALL of the devices are in the 'Copied' state.
```

Example 3: Pulling data online from IBM F20 to VMAX Family array

This example shows how to migrate using a hot pull from an IBM F20 array to a VMAX Family array. Oracle is part of the environment, as is the Veritas volume manager and file system. The example shows how to perform Open Replicator operations on the controlling host connected to the VMAX Family array, and how to perform operations on the remote host connected to the F20 array.

The following examples and tasks show:

1. “Identifying IBM devices” — gets WWNs of the HDS array devices.
2. “Creating device file”
3. “Creating migration session”
4. “Shutdown remote application running on F20 devices”
5. “Activating migration session”
6. “Querying migration session”

7. “Setting ceiling value”
8. “Adjusting ceiling value”

⚠CAUTION

An application on the control devices can be run while Open Replicator is pulling remote data to those devices. The "copy-on-first-access" mechanism is used if the array host reads or writes data on tracks that have not been copied yet from the remote devices. In the case of write I/O, the I/O is temporarily suspended, the track is copied, and then the write is applied to the track. These changed tracks are not reflected back to the remote array.

Prerequisite tasks

Prior to running Open Replicator in this environment, the following tasks must be performed:

- The Fibre Channel switch needs a zone from the array’s FA(s) to the IBM F20 host adapter(s).
- "Hosts" on the IBM F20, that represent the FA(s) on the array need to be configured, and the IBM devices need to be assigned access to those hosts.

Identifying IBM devices

The remote IBM devices that will be migrated to the control-side array must be identified. The EMC Inquiry Utility (version 7.3) can accomplish this when run on the remote host connected to the IBM array. If the Inquiry Utility is not available, use IBM tools. The following `inq` command identifies the IBM storage devices. Open Replicator needs to know the WWN of each IBM device:

```
inq -shark_wwn

For help type inq -h.

----------------------------------------------------------------------
IBM Device          Unit Serial          WWN
----------------------------------------------------------------------
/dev/rdsk/c3t10d0s2 02720499 49424d2020202020323130352020202020202020202020303237323230343939
/dev/rdsk/c3t10d1s2 02820499 49424d2020202020323130352020202020202020202020303238323230343939
/dev/rdsk/c3t10d2s2 02920499 49424d2020202020323130352020202020202020202020303239323230343939
/dev/rdsk/c3t10d3s2 02A20499 49424d2020202020323130352020202020202020202020303241323230343939
/dev/rdsk/c3t10d4s2 02B20499 49424d2020202020323130352020202020202020202020303242323230343939
/dev/rdsk/c3t10d5s2 02C20499 49424d2020202020323130352020202020202020202020303243323230343939
/dev/rdsk/c3t10d6s2 02D20499 49424d2020202020323130352020202020202020202020303244323230343939
/dev/rdsk/c3t10d7s2 02E20499 49424d2020202020323130352020202020202020202020303245323230343939
/dev/rdsk/c3t10d8s2 02F20499 49424d2020202020323130352020202020202020202020303246323230343939
/dev/rdsk/c3t10d9s2 03020499 49424d2020202020323130352020202020202020202020303330323230343939
```

Creating device file

After identifying the control devices (`1C4` – `1CD`) that will receive the data, define the control/remote device pairings in a text file. The following command uses the `vi` text editor to create a text file named `devfile.pull`. The first pairing entered in this file is control device `1C4` on control array `000187990125` (abbreviated as `25`), paired with the

IBM device whose WWN is `49424d2020202020323130352020202020202020202020202030323730343939`. The next control device is paired with the next remote IBM device, and so forth:

```
vi devfile.pull
```

```
SYMDEV=25:1C4  wwn=49424d2020202020323130352020202020202020202020202030323730343939
SYMDEV=25:1C5  wwn=49424d2020202020323130352020202020202020202020202030323830343939
SYMDEV=25:1C6  wwn=49424d2020202020323130352020202020202020202020202030323930343939
SYMDEV=25:1C7  wwn=49424d2020202020323130352020202020202020202020202030324130343939
SYMDEV=25:1C8  wwn=49424d2020202020323130352020202020202020202020202030324230343939
SYMDEV=25:1C9  wwn=49424d2020202020323130352020202020202020202020202030324330343939
SYMDEV=25:1CA  wwn=49424d2020202020323130352020202020202020202020202030324430343939
SYMDEV=25:1CB  wwn=49424d2020202020323130352020202020202020202020202030324530343939
SYMDEV=25:1CC  wwn=49424d2020202020323130352020202020202020202020202030324630343939
SYMDEV=25:1CD  wwn=49424d2020202020323130352020202020202020202020202030333030343939
```

Creating migration session

A `symrcopy create` command from the control host now sets up the Open Replicator hot pull operation. The command creates ten online copy sessions so that data on the remote IBM devices specified in file devfile.pull can be copied to the control devices when the copy operation is started. The `-pull` parameter specifies that the control array is pulling the data to it. The `-hot` parameter indicates that the array application remains online during the operation. The `-name` option gives these sessions the label name IBM:

```
symrcopy create -name IBM -pull -hot -file devfile.pull -noprompt
```

```
'Create' operation execution is in progress for the device list in
device file 'devfile.pull'. Please wait...

'Create' operation successfully executed for the device list
in device file 'devfile.pull'.
```

Shutdown remote application running on F20 devices

Although not shown here, On the remote host, shut down the remote application that uses the F20 array devices, unmount the remote file system(s), and deport volume group(s). By performing these steps after creating the Open Replicator session, this ensures that the create is successful and the setup is correct before incurring application down time.

Activating migration session

The `symrcopy activate` command activates the copy sessions for the pairings in the file devfile.pull. Copying from the remote IBM array to the control-side array begins. At this point the migrated data can be accessed on the control-side array. The copy operation does need to be complete:

```
symrcopy activate -file devfile.pull -noprompt
```

```
'Activate' operation execution is in progress for the device list in
device file 'devfile.pull'. Please wait...

'Activate' operation successfully executed for the device list
in device file 'devfile.pull'.
```

Immediately after a successful activate command and before copy operations are complete, volume group(s) can be imported, file system(s) mounted on the control host, and the application can be run on the VMAX Family array.

Querying migration session

The `symrcopy query` command with the `-detail` option indicates that the sessions for the device pairs defined in the file devfile.pull are in the CopyInProg state and the percent (0%) completion. The display also contains other details such as the pace. The default pace value of 5 provides relatively fast copy time with only a moderate impact on the application:

symrcopy query -file devfile.pull -detail

```
Device File Name      : devfile.pull

Control Device                          Remote Device                  Flags     Status    Done Pace Name
------------------------------ ---------------------------------- ----- -------------- ---- ---- -------
                  Protected Modified
SID:symdev        Tracks   Tracks        Identification               RI CDSHUTZ CTL <=> REM   (%)
----------- --------- --------- ---------------------------------- -- ----- -------------- ---- ---- ------
000187990125:001C4    304380       0 49424D20202020203231303520202 02* .W X..X.S. CopyInProg      0   5 IBM
000187990125:001C5    304382       0 49424D20202020203231303520202 02* .W X..X.S. CopyInProg      0   5 IBM
000187990125:001C6    304383       0 49424D20202020203231303520202 02* .W X..X.S. CopyInProg      0   5 IBM
000187990125:001C7    304384       0 49424D20202020203231303520202 02* .W X..X.S. CopyInProg      0   5 IBM
000187990125:001C8    304387       0 49424D20202020203231303520202 02* .W X..X.S. CopyInProg      0   5 IBM
000187990125:001C9    304387       0 49424D20202020203231303520202 02* .W X..X.S. CopyInProg      0   5 IBM
000187990125:001CA    304389       0 49424D20202020203231303520202 02* .W X..X.S. CopyInProg      0   5 IBM
000187990125:001CB    304390       0 49424D20202020203231303520202 02* .W X..X.S. CopyInProg      0   5 IBM
000187990125:001CC    304391       0 49424D20202020203231303520202 02* .W X..X.S. CopyInProg      0   5 IBM
000187990125:001CD    304391       0 49424D20202020203231303520202 02* .W X..X.S. CopyInProg      0   5 IBM

Total              ---------
  Track(s)           3043864
  MB(s)              95120.8

Legend:
         R (Remote Device Vendor Identification)
         S = Symmetrix, . = Unknown.

         I:  (Remote Device Specification Identifier)
         D = Device Name, W = LUN WWN, World Wide Name.

         Flags:
         (C): X = The background copy setting is active for this pair.
              . = The background copy setting is not active for this pair.
         (D): X = The session is a differential copy session.
              . = The session is not a differential copy session.
         (S): X = The session is pushing data to the remote device(s).
              . = The session is pulling data from the remote device(s).
         (H): X = The session is a hot copy session.
              . = The session is a cold copy session.
         (U): X = The session has donor update enabled.
              . = The session does not have donor update enabled.
         (T): M = The session is a migration session.
              R = The session is a RecoverPoint session.
              S = The session is a standard ORS session.
         (Z): X = The session has front-end zero detection enabled.
              . = The session does not have front-end zero detection enabled.
         (*): The failed session can be reactivated.
```

Note: The Pace column will report N/A for sessions where the control device is on an array running HYPERMAX OS 5977 or higher.

Setting ceiling value

The ceiling value is the percentage of the bandwidth available for Open Replicator background copy transfers. This value can be set but should *only* be done after understanding the bandwidth being used by all other applications. There may be other applications using the same Fibre Channel director(s) as Open Replicator. Setting the

Open Replicator ceiling too high for a director/port can have an adverse impact on these other applications. The ceiling settings that this example uses are for demonstration purposes only.

By default, the ceiling is undefined (as indicated by NONE in the display). The “Max” value is the estimated maximum bandwidth (MB/second) for each director/port of the control-side array. A bandwidth ceiling can be set that balances application performance against Open Replicator copy time. Because the ceiling is not set, the speed of the copy operation is currently controlled by the default pace setting (5) displayed earlier. The following `list ceiling` command to lists the current ceiling setting.

```
symqos -rcopy list ceiling -dir all -sid 25

   Symmetrix ID: 000187990125

       Symmetrix Remote Copy Bandwidth Ceiling

                       Max   Set   Actual
                Dir:P  (MB)  (%)    (MB)
                -----  ----  ----  ------
                01C:0   130  NONE       0
                01C:1   130  NONE       0
                02C:0   130  NONE       0
                02C:1   130  NONE       0
                15C:0   130  NONE       0
                15C:1   130  NONE       0
                16C:0   130  NONE       0
                16C:1   130  NONE       0
                02D:0   130  NONE       0
                02D:1   130  NONE       0
                16D:0   130  NONE       0
                16D:1   130  NONE       0
```

The `symqos -rcopy set ceiling` command sets a bandwidth ceiling of 10% for all director/ports in the VMAX array (`sid 25`). This means that Open Replicator’s ceiling will be 10% of the estimated 130 MB/second FA bandwidth:

```
symqos -rcopy set ceiling 10 -dir all -sid 25 -noprompt

'Set Ceiling' operation execution is in progress

'Set Ceiling' operation successfully executed
```

Adjusting ceiling value

The `symqos -rcopy list ceiling` command shows that the ceiling settings for all director/ports in the VMAX array are now at 10%. Because the control devices are mapped only to director/port `16C:1`, copying occurs only through this director/port. Note that the “Actual” bandwidth being used by Open Replicator for this operation is 13 MB/second, which is 10% of the estimated maximum. The pace value that controlled copy speed earlier is now ignored for any copy session that uses an FA where the ceiling is set:

```
symqos -rcopy list ceiling - dir all -sid 25

Symmetrix ID: 000187990125

    Symmetrix Remote Copy Bandwidth Ceiling

                   Max   Set   Actual
            Dir:P  (MB)  (%)    (MB)
            -----  ----  ----  ------
```

```
01C:0   130    10        0
01C:1   130    10        0
02C:0   130    10        0
02C:1   130    10        0
15C:0   130    10        0
15C:1   130    10        0
16C:0   130    10        0
16C:1   130    10       13
02D:0   130    10        0
02D:1   130    10        0
16D:0   130    10        0
16D:1   130    10        0
```

Another `symrcopy query` command displays the status of the copy operation at 30-second intervals:

symrcopy query -file devfile.pull -detail -i 30

```
Device File Name      : devfile.pull

Control Device                          Remote Device            Flags     Status     Done Pace Name
---------------------------------------- ---------------------------------- ----- ------------- ---- ----
                  Protected Modified
SID:symdev        Tracks    Tracks     Identification                  RI CDSHUTZ CTL <=> REM   (%)
------------------ --------- --------- ------------------------------- -- ----- -------------- ---- ----
000187990125:001C4    299935       0 49424D202020202032313035202020 2* .W X..X.S. CopyInProg      1   5 IBM
000187990125:001C5    299415       0 49424D202020202032313035202020 2* .W X..X.S. CopyInProg      1   5 IBM
000187990125:001C6    299500       0 49424D202020202032313035202020 2* .W X..X.S. CopyInProg      1   5 IBM
000187990125:001C7    299584       0 49424D202020202032313035202020 2* .W X..X.S. CopyInProg      1   5 IBM
000187990125:001C8    299692       0 49424D202020202032313035202020 2* .W X..X.S. CopyInProg      1   5 IBM
000187990125:001C9    299772       0 49424D202020202032313035202020 2* .W X..X.S. CopyInProg      1   5 IBM
000187990125:001CA    299774       0 49424D202020202032313035202020 2* .W X..X.S. CopyInProg      1   5 IBM
000187990125:001CB    299989       0 49424D202020202032313035202020 2* .W X..X.S. CopyInProg      1   5 IBM
000187990125:001CC    300146       0 49424D202020202032313035202020 2* .W X..X.S. CopyInProg      1   5 IBM
000187990125:001CD    301044       0 49424D202020202032313035202020 2* .W X..X.S. CopyInProg      1   5 IBM
Total               ---------
  Track(s)            2998851
  MB(s)               93714.1

Copy rate                      :     13.0 MB/S
Estimated time to completion   : 02:00:11

Legend:
         R (Remote Device Vendor Identification)
         S = Symmetrix, . = Unknown.

         I:  (Remote Device Specification Identifier)
         D = Device Name, W = LUN WWN, World Wide Name.

         Flags:
         (C): X = The background copy setting is active for this pair.
              . = The background copy setting is not active for this pair.
         (D): X = The session is a differential copy session.
              . = The session is not a differential copy session.
         (S): X = The session is pushing data to the remote device(s).
              . = The session is pulling data from the remote device(s).
         (H): X = The session is a hot copy session.
              . = The session is a cold copy session.
         (U): X = The session has donor update enabled.
              . = The session does not have donor update enabled.
         (T): M = The session is a migration session.
              R = The session is a RecoverPoint session.
              S = The session is a standard ORS session.
         (Z): X = The session has front-end zero detection enabled.
              . = The session does not have front-end zero detection enabled.
         (*): The failed session can be reactivated.
```

Note: The Pace column will report N/A for sessions where the control device is on an array running HYPERMAX OS 5977 or higher.

Another `symqos -rcopy set ceiling` command sets a new bandwidth ceiling of 80% for director 16c, port 1, giving Open Replicator most of the possible FA bandwidth. Most likely this setting would impact any applications using director/port FA `16C:1`:

```
symqos -rcopy set ceiling 80 -dir 16c -port 1 -sid 25 -noprompt

'Set Ceiling' operation execution is in progress

'Set Ceiling' operation successfully executed
```

The following `symqos -rcopy list ceiling` command displays the ceiling setting for all directors, including director 16c, port 1. Although the actual bandwidth being used (currently 37 MB/second) is not at 80% of the maximum, it may approach that value as the copy operation progresses. However, the “Actual” value is affected by the SAN and the remote storage, which may keep this value below the percentage allowed for Open Replicator. If the ceiling is never reached, then the ceiling does not affect the copy rate.

```
symqos -rcopy list ceiling -dir all -sid 25


Symmetrix ID: 000187990125

    Symmetrix Remote Copy Bandwidth Ceiling

                   Max   Set   Actual
            Dir:P  (MB)  (%)    (MB)
            -----  ----  ----  ------
            01C:0   130    10       0
            01C:1   130    10       0
            02C:0   130    10       0
            02C:1   130    10       0
            15C:0   130    10       0
            15C:1   130    10       0
            16C:0   130    10       0
            16C:1   130    80      37
            02D:0   130    10       0
            02D:1   130    10       0
            16D:0   130    10       0
            16D:1   130    10       0
```

The `symrcopy verify` command checks at 60-second intervals (`-i`) whether the control/remote device pairs are in the Copied state. The Open Replicator copy operation is now complete:

```
symrcopy verify -copied -file devfile.pull -i 60

NONE of the devices are in the 'Copied' state.

NONE of the devices are in the 'Copied' state.
....
ALL of the devices are in the 'Copied' state.
```

The `symrcopy terminate` command ends all copy sessions defined in the file `devfile.pull`:

```
symrcopy terminate -file devfile.pull -noprompt
```

```
'Terminate' operation execution is in progress for the device list in
device file 'devfile.pull'. Please wait...

'Terminate' operation successfully executed for the device list
in device file 'devfile.pull'.
```

With the copy operation complete, the remote application on the remote host can be restarted (if necessary). However, any changes to remote data at this point are not migrated to the VMAX array unless another full Open Replicator pull operation is performed.

Example 4: Pushing data online from VMAX Family array to Hitachi HDS 9960 array

This example shows how to perform a hot push of data from a Symmetrix VMAX family array to a Hitachi HDS 9960 array. The example shows how to perform Open Replicator operations on the controlling host connected to the VMAX Family array, and how to perform operations on the remote host connected to the HDS array.

The following examples and tasks show:

1. "Identifying HDS devices" — gets WWNs of the HDS array devices.
2. "Creating device file"
3. "Creating migration session"
4. "Setting ceiling value"
5. "Activating migration session"
6. "Adjusting ceiling value"

Note: For arrays running HYPERMAX OS 5977, push operations are not supported.

CAUTION

Applications against remote HDS devices cannot be run at any time during the copy operation.

Prerequisite tasks

Prior to running Open Replicator in this environment, the following tasks must be performed:

1. Configure the Fibre Channel switch to zone the VMAX Family array to the HDS array.
2. Configure the HDS array to assign devices to the VMAX Family array.

Identifying HDS devices

The example needs to identify the remote HDS devices that will receive data from the control-side array. EMC's Inquiry Utility version 7.3 (SIL version 6.0.2) can accomplish this when run on the remote host connected to the HDS array. If the Inquiry Utility is not available, use HDS tools. The following `inq` command identifies the HDS storage devices. Open Replicator needs to know the WWN of each HDS device:

```
inq -hds_wwn
```

```
                        Inquiry utility, Version V7.3-690 (Rev 0.38)
                        (SIL Version V6.0.2.0 (Edit Level 640)
                        Copyright (C) by EMC Corporation, all rights reserved.
                        For help type inq -h.

-------------------------------------------------------------------------------------------
-
HDS Device                Array Serial #   WWN                                        Array Type
-------------------------------------------------------------------------------------------
-
/dev/rdsk/c3t10d0s2       65535            48495441434849205234303039434244303030  R400
/dev/rdsk/c3t10d1s2       65535            48495441434849205234303039434244303032  R400
/dev/rdsk/c3t10d2s2       65535            48495441434849205234303039434244303034  R400
/dev/rdsk/c3t10d3s2       65535            48495441434849205234303039434244303036  R400
/dev/rdsk/c3t10d4s2       65535            48495441434849205234303039434244303038  R400
/dev/rdsk/c3t10d5s2       65535            48495441434849205234303039434244303041  R400
/dev/rdsk/c3t10d6s2       65535            48495441434849205234303039434244303043  R400
/dev/rdsk/c3t10d7s2       65535            48495441434849205234303039434244303045  R400
```

Creating device file

After identifying the control devices (`1C4` – `1CD`) that will send the data, define the control/remote device pairings in a text file. The following command uses the `vi` text editor to create a text file named `devfile.push`. The first pairing entered in this file is control device `1C4` on array `000187990125` (abbreviated as `25`), paired with the HDS device whose WWN is `48495441434849205234303039434244303030`. The next control device is paired with the next remote HDS device, and so forth:

```
vi devfile.push

   symdev=25:1C4  wwn=48495441434849205234303039434244303030
   symdev=25:1C5  wwn=48495441434849205234303039434244303032
   symdev=25:1C6  wwn=48495441434849205234303039434244303034
   symdev=25:1C7  wwn=48495441434849205234303039434244303036
   symdev=25:1C8  wwn=48495441434849205234303039434244303038
   symdev=25:1C9  wwn=48495441434849205234303039434244303041
   symdev=25:1CA  wwn=48495441434849205234303039434244303043
   symdev=25:1CB  wwn=48495441434849205234303039434244303045
   symdev=25:1CC  wwn=48495441434849205234303039434244303130
   symdev=25:1CD  wwn=48495441434849205234303039434244303132
```

Creating migration session

A `symrcopy create` command from the control host now sets up the Open Replicator hot push operation. The command creates ten online copy sessions so that data on the control devices specified in file `devfile.push` can be copied to the remote HDS devices when the copy operation is started. The `-push` parameter specifies that the control array is pushing the data to the remote array. The `-hot` parameter indicates that the array application remains online during the operation. Subsequent copying during these copy sessions will perform incremental copies, capturing only new writes to the control devices. The `-name` option gives these sessions the label name HDS:

```
symrcopy create -name HDS -push -hot -file devfile.push -noprompt

'Create' operation execution is in progress for the device list in
device file 'devfile.push'. Please wait...

'Create' operation successfully executed for the device list
in device file 'devfile.push'.
```

The `symrcopy query` command indicates that the sessions for the device pairs defined in the file `devfile.plus` are in the Created state. To display more detail, include the `-detail` option. To see the full WWN identifier of each remote device, include the `-wwn` option:

symrcopy query -file devfile.push

```
Device File Name      : devfile.push

       Control Device                  Remote Device             Flags     Status     Done
---------------------------- ----------------------------------- ----- -------------- ----
                    Protected
SID:symdev          Tracks     Identification                   RI CDSHUTZ CTL <=> REM     (%)
------------------ --------- -------------------------------- -- ----- -------------- ----
000187990125:001C4     435150 484954414348492052343030394 3424* .W XXXX.S. Created         N/A
000187990125:001C5     435150 484954414348492052343030394 3424* .W XXXX.S. Created         N/A
000187990125:001C6     435150 484954414348492052343030394 3424* .W XXXX.S. Created         N/A
000187990125:001C7     435150 484954414348492052343030394 3424* .W XXXX.S. Created         N/A
000187990125:001C8     435150 484954414348492052343030394 3424* .W XXXX.S. Created         N/A
000187990125:001C9     435150 484954414348492052343030394 3424* .W XXXX.S. Created         N/A
000187990125:001CA     435150 484954414348492052343030394 3424* .W XXXX.S. Created         N/A
000187990125:001CB     435150 484954414348492052343030394 3424* .W XXXX.S. Created         N/A
000187990125:001CC     435150 484954414348492052343030394 3424* .W XXXX.S. Created         N/A
000187990125:001CD     435150 484954414348492052343030394 3424* .W XXXX.S. Created         N/A

Total              ---------
  Track(s)           4351500
  MB(s)               135984

Legend:
         R (Remote Device Vendor Identification)
         S = Symmetrix, . = Unknown.

         I:  (Remote Device Specification Identifier)
         D = Device Name, W = LUN WWN, World Wide Name.

         Flags:
         (C): X = The background copy setting is active for this pair.
              . = The background copy setting is not active for this pair.
         (D): X = The session is a differential copy session.
              . = The session is not a differential copy session.
         (S): X = The session is pushing data to the remote device(s).
              . = The session is pulling data from the remote device(s).
         (H): X = The session is a hot copy session.
              . = The session is a cold copy session.
         (U): X = The session has donor update enabled.
              . = The session does not have donor update enabled.
         (T): M = The session is a migration session.
              R = The session is a RecoverPoint session.
              S = The session is a standard ORS session.
         (Z): X = The session has front-end zero detection enabled.
              . = The session does not have front-end zero detection enabled.
         (*): The failed session can be reactivated.
```

Setting ceiling value

The ceiling value is the percentage of the bandwidth available for Open Replicator background copy transfers. This value can be set but should *only* be done after understanding the bandwidth being used by all other applications. There may be other applications using the same Fibre Channel director(s) as Open Replicator. Setting the Open Replicator ceiling too high for a director/port can have an adverse impact on these other applications. The ceiling settings that this example uses are for demonstration purposes only.

By default, the ceiling is undefined (as indicated by NONE in the display). The “Max” value is the estimated maximum bandwidth (MB/second) for each director/port of the control-side array. A bandwidth ceiling can be set that balances application performance against Open Replicator copy time:

```
symqos -rcopy list ceiling -dir all -sid 25

Symmetrix ID: 000187990125

    Symmetrix Remote Copy Bandwidth Ceiling

                   Max   Set   Actual
            Dir:P  (MB)  (%)    (MB)
            -----  ----  ----  ------
            01C:0   130  NONE       0
            01C:1   130  NONE       0
            02C:0   130  NONE       0
            02C:1   130  NONE       0
            15C:0   130  NONE       0
            15C:1   130  NONE       0
            16C:0   130  NONE       0
            16C:1   130  NONE       0
            02D:0   130  NONE       0
            02D:1   130  NONE       0
            16D:0   130  NONE       0
            16D:1   130  NONE       0
```

The `symqos -rcopy set ceiling` command sets a bandwidth ceiling of 10% for all director/ports in the control-side array (`sid 25`). This means that Open Replicator’s ceiling will be 10% of the estimated 130 MB/second FA bandwidth:

```
symqos -rcopy set ceiling 10 -dir all -sid 25 -noprompt

'Set Ceiling' operation execution is in progress

'Set Ceiling' operation successfully executed
```

The `symqos -rcopy list ceiling` command displays that the ceiling settings for all director/ports in the control-side array are now at 10%. Once the Open Replicator session is activated, the `ceiling` can be displayed again to show its “Actual” value. Note that the pace value (including the default) is ignored for any copy session that uses an FA where the ceiling is set:

```
symqos -rcopy list ceiling -dir all -sid 25

Symmetrix ID: 000187990125

    Symmetrix Remote Copy Bandwidth Ceiling

                   Max   Set   Actual
            Dir:P  (MB)  (%)    (MB)
            -----  ----  ----  ------
            01C:0   130    10       0
            01C:1   130    10       0
            02C:0   130    10       0
            02C:1   130    10       0
            15C:0   130    10       0
            15C:1   130    10       0
            16C:0   130    10       0
            16C:1   130    10       0
            02D:0   130    10       0
            02D:1   130    10       0
            16D:0   130    10       0
            16D:1   130    10       0
```

Activating migration session

The `symrcopy activate` command starts the copy operation for the device pairs defined in the file `devfile.push`. Copying from the control devices to the remote devices begins. Using the `-consistent` option creates a consistent point-in-time copy:

```
symrcopy activate -file devfile.push -consistent -noprompt

'Activate' operation execution is in progress for the device list in
device file 'devfile.push'. Please wait...

'Activate' operation successfully executed for the device list
in device file 'devfile.push'.
```

Adjusting ceiling value

The `symqos -rcopy list ceiling` command shows that the ceiling settings for all director/ports in the control-side array are now at 10%. Because the control devices are mapped only to director/port `16C:1`, copying occurs only through this director/port. Note that the “Actual” bandwidth being used by Open Replicator for this operation is 13 MB/second, which is 10% of the estimated maximum:

```
symqos -rcopy list ceiling -dir all -sid 25

Symmetrix ID: 000187990125

    Symmetrix Remote Copy Bandwidth Ceiling

                   Max   Set   Actual
            Dir:P  (MB)  (%)    (MB)
            -----  ----  ----  ------
            01C:0   130    10       0
            01C:1   130    10       0
            02C:0   130    10       0
            02C:1   130    10       0
            15C:0   130    10       0
            15C:1   130    10       0
            16C:0   130    10       0
            16C:1   130    10      13
            02D:0   130    10       0
            02D:1   130    10       0
            16D:0   130    10       0
            16D:1   130    10       0
```

Note:

The `symrcopy query` command displays the status of the copy operation at 30-second intervals:

```
symrcopy query -file devfile.push -i 30

Device File Name      : devfile.push

       Control Device                  Remote Device             Flags     Status     Done
---------------------------- ----------------------------------- ----- -------------- ----
                     Protected
SID:symdev           Tracks    Identification                   RI CDSHUTZ CTL <=> REM     (%)
------------------ --------- -------------------------------- -- ----- -------------- ----
000187990125:001C4     416645 484954414348492052343030394342 4* .W XXXX.S. CopyInProg     4
000187990125:001C5     416888 484954414348492052343030394342 4* .W XXXX.S. CopyInProg     4
000187990125:001C6     416691 484954414348492052343030394342 4* .W XXXX.S. CopyInProg     4
000187990125:001C7     416632 484954414348492052343030394342 4* .W XXXX.S. CopyInProg     4
000187990125:001C8     416799 484954414348492052343030394342 4* .W XXXX.S. CopyInProg     4
```

```
000187990125:001C9     417123 484954414348492052343030394342 4* .W XXXX.S. CopyInProg     4
000187990125:001CA     416876 484954414348492052343030394342 4* .W XXXX.S. CopyInProg     4
000187990125:001CB     417092 484954414348492052343030394342 4* .W XXXX.S. CopyInProg     4
000187990125:001CC     417009 484954414348492052343030394342 4* .W XXXX.S. CopyInProg     4
000187990125:001CD     416717 484954414348492052343030394342 4* .W XXXX.S. CopyInProg     4

Total              ---------
  Track(s)           4168472
  MB(s)               130265

Legend:
        R (Remote Device Vendor Identification)
        S = Symmetrix, C = Clariion, . = Unknown.

        I:  (Remote Device Specification Identifier)
        D = Device Name, W = LUN WWN, World Wide Name.

        Flags:
        (C): X = The background copy setting is active for this pair.
             . = The background copy setting is not active for this pair.
        (D): X = The session is a differential copy session.
             . = The session is not a differential copy session.
        (S): X = The session is pushing data to the remote device(s).
             . = The session is pulling data from the remote device(s).
        (H): X = The session is a hot copy session.
             . = The session is a cold copy session.
        (U): X = The session has donor update enabled.
             . = The session does not have donor update enabled.
        (T): M = The session is a migration session.
             R = The session is a RecoverPoint session.
             S = The session is a standard ORS session.
        (Z): X = The session has front-end zero detection enabled.
             . = The session does not have front-end zero detection enabled.
        (*): The failed session can be reactivated.
```

The `symqos -rcopy set ceiling` command sets a bandwidth ceiling of 100% for all director/ports in the control-side array (`sid 25`), giving Open Replicator all of the possible FA bandwidth. Most likely this setting would impact any applications using director/port FA 16C:1:

```
symqos -rcopy set ceiling 100 -dir all -sid 25 -noprompt
```

```
'Set Ceiling' operation execution is in progress.

'Set Ceiling' operation successfully executed.
```

The following `symqos -rcopy` command displays the new ceiling setting for all directors, including director 16c, port 1. Although the actual bandwidth being used (currently 47 MB/second) is not at 100% of the maximum, it may approach that value as the copy operation progresses. However, the “Actual” value is affected by the SAN and the remote storage, which may keep this value below the estimated maximum of the control-side array director/port. If the ceiling is never reached, then the ceiling does not affect the copy rate:

```
symqos -rcopy list ceiling -dir all -sid 25
```

```
Symmetrix ID: 000187990125

    Symmetrix Remote Copy Bandwidth Ceiling

                   Max   Set   Actual
            Dir:P  (MB)  (%)    (MB)
            -----  ----  ----  ------
```

```
                                    01C:0   130   100       0
                                    01C:1   130   100       0
                                    02C:0   130   100       0
                                    02C:1   130   100       0
                                    15C:0   130   100       0
                                    15C:1   130   100       0
                                    16C:0   130   100       0
                                    16C:1   130   100      47
                                    02D:0   130   100       0
                                    02D:1   130   100       0
                                    16D:0   130   100       0
                                    16D:1   130   100       0
```

Another `symrcopy query` command displays an updated status of the copy operation at 30-second intervals:

```
symrcopy query -file devfile.push -i 30

Device File Name      : devfile.push

       Control Device                  Remote Device             Flags     Status     Done
---------------------------- ----------------------------------- ----- -------------- ----
                   Protected
SID:symdev         Tracks     Identification                    RI CDSHUTZ CTL <=> REM     (%)
------------------ --------- ---------------------------------- -- ----- -------------- ----
000187990125:001C4     390617 484954414348492052343030394342 4* .W XXXX.S. CopyInProg       10
000187990125:001C5     381968 484954414348492052343030394342 4* .W XXXX.S. CopyInProg       12
000187990125:001C6     386389 484954414348492052343030394342 4* .W XXXX.S. CopyInProg       11
000187990125:001C7     386463 484954414348492052343030394342 4* .W XXXX.S. CopyInProg       11
000187990125:001C8     381195 484954414348492052343030394342 4* .W XXXX.S. CopyInProg       12
000187990125:001C9     399176 484954414348492052343030394342 4* .W XXXX.S. CopyInProg        8
000187990125:001CA     396252 484954414348492052343030394342 4* .W XXXX.S. CopyInProg        8
000187990125:001CB     399542 484954414348492052343030394342 4* .W XXXX.S. CopyInProg        8
000187990125:001CC     398678 484954414348492052343030394342 4* .W XXXX.S. CopyInProg        8
000187990125:001CD     397450 484954414348492052343030394342 4* .W XXXX.S. CopyInProg        8

Total              ---------
  Track(s)           3917730
  MB(s)               122429

Copy rate                      : 48.3 MB/S
Estimated time to completion   : 00:42:16

Legend:
         R (Remote Device Vendor Identification)
         S = Symmetrix, . = Unknown.

         I:  (Remote Device Specification Identifier)
         D = Device Name, W = LUN WWN, World Wide Name.

         Flags:
         (C): X = The background copy setting is active for this pair.
              . = The background copy setting is not active for this pair.
         (D): X = The session is a differential copy session.
              . = The session is not a differential copy session.
         (S): X = The session is pushing data to the remote device(s).
              . = The session is pulling data from the remote device(s).
         (H): X = The session is a hot copy session.
              . = The session is a cold copy session.
         (U): X = The session has donor update enabled.
              . = The session does not have donor update enabled.
         (T): M = The session is a migration session.
              R = The session is a RecoverPoint session.
              S = The session is a standard ORS session.
         (Z): X = The session has front-end zero detection enabled.
              . = The session does not have front-end zero detection enabled.
```

```
(*): The failed session can be reactivated.
```

The `symrcopy verify` command checks at 60-second intervals (`-i`) to verify whether the control/remote device pairs are in the Copied state. The Open Replicator copy operation is now complete:

```
symrcopy verify -Copied -i 60 -file devfile.push

NONE of the devices are in the 'Copied' state.

NONE of the devices are in the 'Copied' state.
...

NOT ALL of the devices are in the 'Copied' state.

ALL of the devices are in the 'Copied' state.
```

Example 5: Performing a cold push from a VDEV

The following example shows how to perform a cold push session from a Virtual Device (VDEV) and includes the following steps:

1. "Creating Snap session"
2. "Creating migration session"
3. "Activating migration session"
4. "Querying migration session"

Note: For arrays running HYPERMAX OS 5977, creating ORS sessions on control devices that are involved in a TimeFinder session are not supported. In addition, push operations are not supported.

Creating Snap session

The first step is to create a TimeFinder/Snap session using the `symsnap create` command. Do not activate the session:

Note: Detailed instructions for creating a TimeFinder/Snap session and performing snap operations are provided in the *EMC Solutions Enabler TimeFinder Family CLI Product Guide*.

```
symsnap -g snap1 create -noprompt -v

'Create' operation execution is in progress for
device group 'snap1'. Please wait...


SELECTING Source devices in the group:

  Device: 00E5 [SELECTED]
  Device: 00E9 [SELECTED]

SELECTING Target devices in the group:

  Device: 00AA [SELECTED]
  Device: 00AE [SELECTED]

PAIRING of Source and Target devices:
```

```
  Devices: 00E5(S) - 00AA(T) [(M) PAIRED]
  Devices: 00E6(S) - 00AB(T) [(m) PAIRED]
  Devices: 00E7(S) - 00AC(T) [(m) PAIRED]
  Devices: 00E8(S) - 00AD(T) [(m) PAIRED]
  Devices: 00E9(S) - 00AE(T) [(M) PAIRED]
  Devices: 00EA(S) - 00AF(T) [(m) PAIRED]
  Devices: 00EB(S) - 00B0(T) [(m) PAIRED]
  Devices: 00EC(S) - 00B1(T) [(m) PAIRED]

STARTING a Snap 'CREATE' operation.

The Snap 'CREATE' operation SUCCEEDED.

'Create' operation successfully executed for device group
'snap1'.
```

Verify that the TimeFinder/Snap session is in the created state:

```
symsnap -g snap1 verify -created -i 30
```

```
All devices in the group 'snap1' are in 'Created' state.
```

Perform a query operation on the device group:

```
symsnap -g snap1 query
```

```
Device Group (DG) Name: snap1
DG's Type             : REGULAR
DG's Symmetrix ID     : 000192600161

      Source Device          Target Device                  State     Copy
------------------------- ------------------- ---------- ------------ ----
                Protected                        Changed
Logical   Sym   Tracks    Logical   Sym     G    Tracks  SRC <=> TGT  (%)
------------------------- ------------------- ---------- ------------ ----

DEV001    000E5     552360 VDEV001   000AA    X          0 Created     0
DEV002    000E9     552360 VDEV002   000AE    X          0 Created     0

Total            --------                     ----------
  Track(s)        1104720                             0
  MB(s)           69045.0                           0.0

Legend:

(G): X = The Target device is associated with this group,
     . = The Target device is not associated with this group.
```

Creating migration session

Next, create an Open Replicator session using the `symrcopy create` command to define the snap device as the control device. Do not activate the session. This creates an Open Replicator session on the VDEV. The contents of the device file `vORS_to_symm` for this example is:

```
symdev=000192600161:aa symdev=000192600141:a5d
symdev=000192600161:ae symdev=000192600141:a61
```

```
symrcopy -file vORS_to_symm create -cold -push -noprompt -v
```

```
'Create' operation execution is in progress for the device list
in device file 'vORS_to_symm'. Please wait...
```

```
STARTING a REMOTE Copy CREATE (PUSH) (COLD) (DIFFERENTIAL)

SELECTING Control device - Remote devices:

  (Ctl)Sym: 000192600161 Device: 000AA -   LUN WWN: 60000970000192600141533030413544
   - [SELECTED]
  (Ctl)Sym: 000192600161 Device: 000AE -   LUN WWN: 60000970000192600141533030413631
   - [SELECTED]

STARTING a RCOPY 'CREATE' operation.

SELECTING Control device - Remote devices:

  (Ctl)Sym: 000192600161 Device: 000AA -   LUN WWN: 60000970000192600141533030413544
   - [CREATED]
  (Ctl)Sym: 000192600161 Device: 000AE -   LUN WWN: 60000970000192600141533030413631
   - [CREATED]
The Rcopy 'CREATE' operation SUCCEEDED.
'Create' operation successfully executed for the device list
in device file 'vORS_to_symm'.
```

The `symrcopy query` command indicates that the sessions for the device pairs defined in the file `vORS_to_symm` are in the Created state. To display more detail, include the `-detail` option.

symrcopy -file vORS_to_symm query

```
Device File Name      : vORS_to_symm

       Control Device                  Remote Device             Flags     Status     Done
---------------------------- ---------------------------------- ----- -------------- ----
                    Protected
SID:symdev          Tracks     Identification                   RI CDSHUTZ CTL <=> REM    (%)
------------------ --------- -------------------------------- -- ----- -------------- ----
000192600161:000AA     552360 000192600141:0A5D                SD XXX..S. Created         N/A
000192600161:000AE     552360 000192600141:0A61                SD XXX..S. Created         N/A

Total              ---------
  Track(s)           1104720
  MB(s)              69045.0

Legend:
         R (Remote Device Vendor Identification)
         S = Symmetrix, . = Unknown.

         I:  (Remote Device Specification Identifier)
         D = Device Name, W = LUN WWN, World Wide Name.

         Flags:
         (C): X = The background copy setting is active for this pair.
              . = The background copy setting is not active for this pair.
         (D): X = The session is a differential copy session.
              . = The session is not a differential copy session.
         (S): X = The session is pushing data to the remote device(s).
              . = The session is pulling data from the remote device(s).
         (H): X = The session is a hot copy session.
              . = The session is a cold copy session.
         (U): X = The session has donor update enabled.
              . = The session does not have donor update enabled.
         (T): M = The session is a migration session.
              R = The session is a RecoverPoint session.
              S = The session is a standard ORS session.
         (Z): X = The session has front-end zero detection enabled.
              . = The session does not have front-end zero detection enabled.
         (*): The failed session can be reactivated.
```

Activating Snap session

Activate the Snap session that was created previously using the `symsnap activate` command. If consistency is required, the command will run within a consistency window:

```
symsnap -g snap1 activate -consistent -not_ready -noprompt
```

```
'Activate' operation execution is in progress for
device group 'snap1'. Please wait...

'Activate' operation successfully executed for device group
'snap1'.
```

Perform a query operation on the device group:

```
symsnap -g snap1 query
Device Group (DG) Name: snap1
DG's Type             : REGULAR
DG's Symmetrix ID     : 000192600161


      Source Device          Target Device                  State     Copy
------------------------ ------------------- ---------- ------------ ----
                Protected                          Changed
Logical   Sym   Tracks    Logical   Sym      G    Tracks  SRC <=> TGT  (%)
------------------------ ------------------- ---------- ------------ ----

DEV001    000E5     550904 VDEV001   000AA    X       1456 CopyOnWrite  0
DEV002    000E9     550913 VDEV002   000AE    X       1447 CopyOnWrite  0
0
Total            --------                     ----------
  Track(s)        1101817                          2903
  MB(s)           68863.6                         181.4

Legend:

(G): X = The Target device is associated with this group,
     . = The Target device is not associated with this group.
```

Activating migration session

Activate the Open Replicator session that was created previously using the `symrcopy activate` command. The Open Replicator session begins copying the data:

```
symrcopy -file vORS_to_symm activate -noprompt -v
```

```
'Activate' operation execution is in progress for the device list
in device file 'vORS_to_symm'. Please wait...

STARTING a REMOTE Copy ACTIVATE

SELECTING Control device - Remote devices:

  (Ctl)Sym: 000192600161 Device: 000AA -   LUN WWN: 60000970000192600141533030413544
   - [SELECTED]
  (Ctl)Sym: 000192600161 Device: 000AE -   LUN WWN: 60000970000192600141533030413631
   - [SELECTED]

STARTING a RCOPY 'ACTIVATE' operation.

SELECTING Control device - Remote devices:

  (Ctl)Sym: 000192600161 Device: 000AA -   LUN WWN: 60000970000192600141533030413544
   - [ACTIVATED]
  (Ctl)Sym: 000192600161 Device: 000AE -   LUN WWN: 60000970000192600141533030413631
   - [ACTIVATED]
```

```
The Rcopy 'ACTIVATE' operation SUCCEEDED.
'Activate' operation successfully executed for the device list
in device file 'vORS_to_symm'.
```

Querying migration session

The `symrcopy query` command indicates that the sessions for the device pairs defined in the file `vORS_to_symm` are in the CopyInProg state. To display more detail, include the `-detail` option.

symrcopy -file vORS_to_symm query

```
Device File Name      : vORS_to_symm

       Control Device                  Remote Device             Flags     Status     Done
---------------------------- ---------------------------------- ----- -------------- ----
                   Protected
SID:symdev         Tracks     Identification                   RI CDSHUTZ CTL <=> REM    (%)
------------------ --------- ------------------------------- -- ----- -------------- ----
000192600161:000AA     544488 000192600141:0A5D               SD XXX..S. CopyInProg        1
000192600161:000AE     544481 000192600141:0A61               SD XXX..S. CopyInProg        1

Total              ---------
  Track(s)           1088969
  MB(s)              68060.6

Legend:
         R (Remote Device Vendor Identification)
         S = Symmetrix, . = Unknown.

         I:  (Remote Device Specification Identifier)
         D = Device Name, W = LUN WWN, World Wide Name.

         Flags:
         (C): X = The background copy setting is active for this pair.
              . = The background copy setting is not active for this pair.
         (D): X = The session is a differential copy session.
              . = The session is not a differential copy session.
         (S): X = The session is pushing data to the remote device(s).
              . = The session is pulling data from the remote device(s).
         (H): X = The session is a hot copy session.
              . = The session is a cold copy session.
         (U): X = The session has donor update enabled.
              . = The session does not have donor update enabled.
         (T): M = The session is a migration session.
              R = The session is a RecoverPoint session.
              S = The session is a standard ORS session.
         (Z): X = The session has front-end zero detection enabled.
              . = The session does not have front-end zero detection enabled.
         (*): The failed session can be reactivated.
```

This output shows the copy status as currently copy-in-progress. When copying has finished, terminate the Open Replicator session using the `symrcopy terminate` command. Then terminate the Time Finder/Snap session using the `symsnap terminate` command.

Note: Before terminating, sessions may also be recreated using the `symrcopy recreate` and `symsnap recreate` commands.

CHAPTER 24
Operational Restrictions and State Reference

This chapter details which ORS operations are permissible for certain control device states, devices in various states of a replication (TimeFinder or SRDF) operation, and certain device types.

- Device control operations allowed for ORS control devices 664
- ORS operations allowed for replication session states..... 665
- ORS operations allowed for device types..... 676

Device control operations allowed for ORS control devices

Table 42 details whether a device control operation is permissible, for the ORS control device, in various states with the controlling host (such as, ready, not ready).

Table 42 Device control operations allowed for ORS control device states with host

Action	Allowed
Ready	
Online RCopy control device pushing out	Yes[a]
Online RCopy control device pulling in	Yes
Offline RCopy control device pushing out	No
Offline RCopy control device pulling in	No
Not Ready	
Online RCopy control device pushing out	Yes[a]
Online RCopy control device pulling in	Yes
Offline RCopy control device pushing out	N/A
Offline RCopy control device pulling in	N/A
RW Enable	
Online RCopy control device pushing out	Yes[a]
Online RCopy control device pulling in	Yes
Offline RCopy control device pushing out	No
Offline RCopy control device pulling in	No
Write Disable	
Online RCopy control device pushing out	Yes[a]
Online RCopy control device pulling in	Yes
Offline RCopy control device pushing out	No
Offline RCopy control device pulling in	No

a. Not supported for arrays running HYPERMAX OS 5977 or higher.

ORS operations allowed for replication session states

This section details whether or not an ORS operation is permissible on devices in various states of a replication (TimeFinder or SRDF) session.

TimeFinder/Mirror and TimeFinder/Clone sessions)

The following tables detail whether or not an ORS operation is supported on devices (BCV, STD) in various states of TimeFinder sessions.

- "TimeFinder/Mirror source device is control device for rcopy push session"
- "TimeFinder/Mirror target device is control device for rcopy push session"
- "TimeFinder/Mirror source device is control device for rcopy pull session"
- "TimeFinder/Mirror target device is control device for rcopy pull session"
- "TimeFinder/Clone source device is control device for rcopy push session"
- "TimeFinder/Clone target device is control device for rcopy push session"
- "TimeFinder/Clone source device is control device for rcopy pull session"
- "TimeFinder/Clone target device is control device for rcopy pull session"

Table 43 TimeFinder/Mirror source device is control device for rcopy push session

	BCV Pair State										
rcopy Operation	Never Established	Sync in prog	Synchronized	Split in prog	Split before sync	Split	Split no incremental	Restore in prog	Restored	Split before restore	Invalid
Create	Y[a]	Y[a]	Y[a]	Y[a]	Y[a]	Y[a]	Y[a]		Y[a]		
Activate Recreate Set Precopy	Y	Y	Y	Y	Y	Y	Y		Y		
Incremental Restore	Y	Y	Y	Y	Y	Y	Y		Y	Y	
Set Nocopy Set Copy Set Pace Terminate Remove Rename	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

a. Not supported for arrays running HYPERMAX OS 5977 or higher.

Table 44 TimeFinder/Mirror target device is control device for rcopy push session

	BCV Pair State											
rcopy Operation	Never established	Sync in prog	Synchronized	Split in prog	Background split in progress	Split before sync	Split	Split no incremental	Restore in prog	Restored	Split before restore	Invalid
Create	Y[a]						Y[a]	Y[a]				
Activate Recreate Set Precopy	Y						Y	Y				
Incremental Restore	Y					Y	Y	Y			Y	
Set Nocopy Set Copy Set Pace Terminate Remove Rename	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

a. Not supported for arrays running HYPERMAX OS 5977 or higher.

Table 45 TimeFinder/Mirror source device is control device for rcopy pull session

	BCV Pair State											
rcopy Operation	Never established	Sync in prog	Synchronized	Split in prog	Backgrounds plit in progress	Split before sync	Split	Split no incremental	Restore in prog	Restored	Split before restore	Invalid
Create	Y	Y[a]	Y[a]	Y[a]	Y[a]	Y[a]	Y[a]	Y[a]		Y[a]	Y[a]	
Activate	Y	Y	Y	Y	Y	Y	Y	Y		Y	Y	
Set Nocopy Set Copy Set Pace Terminate Remove Rename Donor Update Off Failback FrontEnd Zero Off	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

a. Not supported for arrays running HYPERMAX OS 5977 or higher.

Table 46 TimeFinder/Mirror target device is control device for rcopy pull session

	BCV Pair State											
rcopy Operation	Never stablished	Sync in prog	Synchronized	Split in prog	Background split in progress	Split before sync	Split	Split no incremental	Restore in prog	Restored	Split before restore	Invalid
Create	Y					Y[a]	Y[a]	Y[a]			Y[a]	
Activate	Y					Y	Y	Y			Y	
Set Nocopy Set Copy Set Pace Terminate Remove Rename Donor Update Off Failback FrontEnd Zero Off	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

a. Not supported for arrays running HYPERMAX OS 5977 or higher.

Table 47 TimeFinder/Clone source device is control device for rcopy push session

	Pair State														
rcopy Operation	No session	Create in prog	Created	Recreated	Precopy	Copy in prog	Copied	Copy on write	Copy on access	Split	Restore in prog	Restored	Terminate in prog	Invalid	Failed
Create	Y[a]	Y [abc]	Y[abc]	Y[abc]	Y[abc]	Y[abc]	Y[abc]	Y[abc]	Y[abc]	Y[abc]	Y[ac]	Y[abc]	Y[ac]	Y[ac]	Y[ac]
Activate Recreate SetPrecopy	Y[a]	Y[ab]	Y[ab]	Y [ab]	Y[ab]	Y[ab]	Y[ab]	Y[ab]	Y[ab]	Y[ab]	Y[a]	Y[ab]	Y[a]	Y[a]	Y[a]
Incremental Restore	Y[a]	Y [abd]	Y[abd]	Y[abd]	Y[abd]	Y[abd]	Y[abd]	Y[abd]	Y[abd]	Y[abd]		Y[abd]			
Set Pace Remove	Y[a]	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Set Nocopy Set Copy Terminate Rename	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

a. Not supported if device is held.

b. Not supported if device is the source of an extent-level Clone session.
c. Not supported for arrays running HYPERMAX OS 5977 or higher.
d. Not supported if device is the source of a Snap or VP Snap session.

Table 48 TimeFinder/Clone target device is control device for rcopy push session

	Pair State														
rcopy Operation	No session	Create in prog	Created	Recreated	Precopy	Copy in prog	Copied	Copy on write	Copy on access	Split	Restore in prog	Restored	Terminate in prog	Invalid	Failed
Create	Y[a]						Y[abcd]			Y[abcd]					
Activate Recreate SetPrecopy Incremental Restore	Y[a]						Y[abd]			Y[abd]					
Set Nocopy Set Copy Set Pace Terminate Remove Rename	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

a. Not supported if device is held.
b. Not supported if device is the source of an extent-level Clone session.
c. Not supported for arrays running HYPERMAX OS 5977 or higher.
d. Not supported if device is the source of a Snap or VP Snap session.

Table 49 TimeFinder/Clone source device is control device for rcopy pull session

	Pair State														
rcopy Operation	No session	Create in prog	Created	Recreated	Precopy	Copy in prog	Copied	Copy on write	Copy on access	Split	Restore in prog	Restored	Terminate in prog	Invalid	Failed
Create	Y[a]	Y[abcd]	Y[abcd]	Y[abcd]	Y[abcd]	Y[abcd]	Y[abcd]	Y[abcd]	Y[abcd]	Y[abcd]		Y[abcd]			
Activate	Y[a]	Y[abd]	Y[abd]	Y[abd]	Y[abd]	Y[abd]	Y[abd]	Y[abd]	Y[abd]	Y[abd]		Y[abd]			
Set Nocopy Set Copy Set Pace Terminate Remove Rename Donor Update Off Failback FrontEnd Zero Off	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

a. Not supported if device is held.
b. Not supported if device is the source of an extent-level Clone session.
c. Not supported for arrays running HYPERMAX OS 5977 or higher.
d. Not supported if device is the source of a Snap or VP Snap session.

Table 50 TimeFinder/Clone target device is control device for rcopy pull session

	Pair State														
rcopy Operation	No session	Create in prog	Created	Recreated	Precopy	Copy in prog	Copied	Copy on write	Copy on access	Split	Restore in prog	Restored	Terminate in prog	Invalid	Failed
Create	Y[a]						Y[abcd]			Y[abcd]					
Activate	Y[a]						Y[abd]			Y[abd]					
Set Nocopy Set Copy Set Pace Terminate Remove Rename Donor Update Off Failback FrontEnd Zero Off	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

a. Not supported if device is held.
b. Not supported if device is the source of an extent-level Clone session.
c. Not supported for arrays running HYPERMAX OS 5977 or higher.
d. Not supported if device is the source of a Snap or VP Snap session.

TimeFinder/SnapVX sessions (HYPERMAX OS 5977)

The following tables detail whether or not an ORS operation is supported on devices in various states of a TimeFinder/SnapVX session:

- "TimeFinder/Snap VX source device is control device for rcopy push session"
- "TimeFinder/SnapVX target device is control device for rcopy push session"
- "TimeFinder/Snap VXsource device is control device for rcopy pull session"
- "TimeFinder/SnapVXtarget device is control device for rcopy pull session"

Table 51 TimeFinder/Snap VX source device is control device for rcopy push session

	TimeFinder/SnapVX State									
rcopy Operation	No session	Establish in prog	Established	Restore in prog	Restored	Terminate in prog	Failed	Link copy in prog	Link copied	Linked
Create	Y[a]							N/A	N/A	N/A
Activate Recreate Set Precopy	Y[a]	Y[a]	Y[a]	Y[a]	Y[a]	Y[a]	Y[a]	N/A	N/A	N/A
Incremental Restore	Y[a]									
Set Nocopy Set Copy Set Pace Terminate Remove Rename	Y	Y	Y	Y	Y	Y	Y	N/A	N/A	N/A

a. Not supported if device is held.

Table 52 TimeFinder/SnapVX target device is control device for rcopy push session

	TimeFinder/SnapVX State									
RCOPY Operation	No session	Establish in prog	Established	Restore in prog	Restored	Terminate in prog	Failed	Link copy in prog	Link copied	Linked
Create	Y[a]	N/A	N/A	N/A	N/A	N/A				
Activate Recreate Set Precopy	Y[a]	N/A	N/A	N/A	N/A	N/A			Y[a]	
Incremental Restore	Y[a]	N/A	N/A	N/A	N/A	N/A				
Set Nocopy Set Copy Set Pace Terminate Remove Rename	Y	NA	NA	NA	NA	NA	Y	Y	Y	

a. Not supported if device is held.

Table 53 TimeFinder/Snap VXsource device is control device for rcopy pull session

	TimeFinder/SnapVX State									
RCOPY Operation	No session	Establish in prog	Established	Restore in prog	Restored	Terminate in prog	Failed	Link copy in prog	Link copied	Linked
Create	Y[a]							N/A	N/A	N/A
Activate	Y[a]	Y[a]	Y[a]		Y[a]			N/A	N/A	N/A
Set Nocopy Set Copy Set Pace Terminate Remove Rename Donor Update Off Failback FrontEnd Zero Off	Y	Y	Y	Y	Y	Y	Y	N/A	N/A	N/A

a. Not supported if device is held.

Table 54 TimeFinder/SnapVXtarget device is control device for rcopy pull session

	TimeFinder/SnapVX State									
RCOPY Operation	No session	Establish in prog	Established	Restore in prog	Restored	Terminate in prog	Failed	Link copy in prog	Link copied	Linked
Create	Y[a]	N/A	N/A	N/A	N/A	N/A				
Activate	Y[a]	N/A	N/A	N/A	N/A	N/A			Y[a]	
Set Nocopy Set Copy Set Pace Terminate Remove Rename Donor Update Off Failback FrontEnd Zero Off	Y	N/A	N/A	N/A	N/A	N/A	Y	Y	Y	

a. Not supported if device is held.

SRDF sessions

The following tables detail whether or not an ORS operation is permissible on devices (such as R1 and R2) in various states of SRDF sessions.

- "SRDF R1 mirror is control device for rcopy push session"
- "SRDF R2 mirror is control device for rcopy push session"
- "SRDF R1 mirror is control device for rcopy pull session"
- "SRDF R2 mirror is control device for rcopy pull session"

Note: If the device is a cascaded SRDF R21 device, then both the R1 and R2 rules listed apply. If the action is not allowed for either an R1 or R2 device, then the action is not allowed for an R21 device.

Table 55 SRDF R1 mirror is control device for rcopy push session

	RDF Mirror State										
RCOPY Operation	Sync in prog	Synchronized	Split	Suspended	Failed Over	Partitioned	R1 updated	R1 update in prog	Invalid	Consistent	Transmit idle
Create	Y[ab]	Y[a]	Y[a]	Y[a]		Y[a]				Y[ab]	Y[ab]
Activate Recreate Set Precopy	Y[b]	Y	Y	Y		Y				Y[b]	Y[b]
Incremental Restore	Y[bc]	Y	Y	Y		Y				Y[bc]	
Set Nocopy Set Copy Set Pace Terminate Remove Rename	Yes for all mirror states listed above										

a. Not supported for arrays running HYPERMAX OS 5977 or higher.
b. Must have no local invalids on R1 or remote invalids on R2.
c. Array must be running Enginuity 5874 or higher.

Table 56 SRDF R2 mirror is control device for rcopy push session

	RDF Mirror State										
RCOPY Operation	Sync in prog	Synchronized	Split	Suspended	Failed over	Partitioned	R1 updated	R1 update in prog	Invalid	Consistent	Transmit idle
Create			Y[a]	Y[a]	Y[a]	Y[a]	Y[a]	Y[a]			
Activate Recreate Set Precopy			Y	Y	Y	Y	Y	Y			
Incremental Restore			Y	Y	Y	Y	Y[b]				
Set Nocopy Set Copy Set Pace Terminate Remove Rename	Yes for all mirror states listed above										

a. Not supported for arrays running HYPERMAX OS 5977 or higher.
b. R2 must be write enabled.

Table 57 SRDF R1 mirror is control device for rcopy pull session

	RDF Mirror State										
RCOPY Operation	Sync in prog	Synchronized	Split	Suspended	Failed over	Partitioned	R1 updated	R1 update in prog	Invalid	Consistent	Transmit idle
Create	Y[abcde]	Y[abcde]	Y[e]	Y[e]		Y[e]				Y[bcde]	
Activate	Y[abcd]	Y[abcd]	Y	Y		Y				Y[abcd]	
Set Nocopy Set Copy Set Pace Terminate Remove Rename Donor Update Off Failback Frontend Zero Off	Yes for all mirror states listed above										

a. Array must be running Enginuity 5874 or higher.
b. Must have no local invalids on R1 or remote invalids on R2.
c. Frontend zero detect must be in Off state.
d. Force flag must be set.
e. Not supported for arrays running HYPERMAX OS 5977 or higher.

Table 58 SRDF R2 mirror is control device for rcopy pull session

	RDF Mirror State										
RCOPY Operation	Sync in prog	Synchronized	Split	Suspended	Failed over	Partitioned	R1 updated	R1 update in prog	Invalid	Consistent	Transmit idle
Create			Y[a]	Y[a]	Y[a]	Y[a]	Y[abc]				
Activate			Y	Y	Y	Y	Y[bc]				
Set Nocopy Set Copy Set Pace Terminate Remove Rename Donor Update Off Failback Frontend Zero Off	Yes for all mirror states listed above										

a. Not supported for arrays running HYPERMAX OS 5977 or higher.

b. R2 must be write enabled.

c. Frontend zero detect must be in Off state.

ORS operations allowed for device types

Table 59 details whether or not an ORS operation is permissible for certain device types.

Note: Open Replicator fully supports copy operations for thin devices. For information on Virtual Provisioning and creating thin devices, refer to the *EMC Solutions Enabler Array Management CLI Product Guide*.

Table 59 ORS operations allowed by device type (page 1 of 2)

Action	Allowed
Rcopy Create push where local device type is:	
Gatekeeper	Yes, as long as it is not the gatekeeper for the syscall.
WORM	No
CKD_3380 CKD_3390	No
AS400	No
Rcopy Create pull, where local device type is:	
Gatekeeper	Yes, as long as it is not the gatekeeper for the syscall.
WORM	No
CKD_3380 CKD_3390	No
AS400	Yes on DMX arrays running Enginuity 5773.150 or higher, and VMAX Family arrays running Enginuity 5874 or higher.
Rcopy Create push or pull, where local device type is:	
Virtual device (VDEV)	Yes, as long as the following conditional are met: • Session is a cold push ONLY • VMAX Family arrays are running Enginuity 5874 through 5876.
SFS device	No
STAR	Yes
Unconfigured device	No
Meta member	No
Rcopy Create push, where remote device type is:	
WORM	No
CKD_3380 CKD_3390	No

Table 59 ORS operations allowed by device type (page 2 of 2)

Action	Allowed
Rcopy Create pull, where remote device type is:	
WORM	No
CKD_3380 CKD_3390	No
AS400	Yes

CHAPTER 25
Federated Live Migration Operations (Enginuity 5876 or lower)

This chapter introduces EMC Federated Live Migration and explains how to use the `symrcopy` command to copy device data, without host disruption, from DMX and VMAX 20K arrays to VMAX 10K, 20K, and 40K arrays in the storage network.

- Federated Live Migration overview 680
- The FLM process 681
- Environmental and system requirements for FLM 682
- Using FLM to perform a migration 684

Federated Live Migration overview

Federated Live Migration (FLM) is used to assist with data center consolidation or technology refresh. FLM copies devices from donor DMX and VMAX 20K arrays to VMAX 10K, 20K and 40K arrays, without needing to restart the application hosts. It does this by having the new array device assume the identity and geometry of the donor array device and then perform a hot pull operation.

IMPORTANT

FLM is not supported on arrays running HYPERMAX OS 5977.

Note: In this document, the use of "donor array(s)" refers to DMX and VMAX 20K arrays and "new array(s)" refers to VMAX 10K, 20K and 40K arrays.

The SYMCLI interface for FLM performs all of the actual data migration; however, it does not automatically setup any of the required zones for the migration. This includes the zones from the application hosts to the new array, and from the donor array to the new array.

⚠CAUTION

Data migrations are often complex operations and require careful planning and execution of predetermined procedures. Failure to identify and perform necessary steps or actions within supported configurations can result in data unavailability or loss. Before proceeding with FLM, EMC strongly recommends that the *Symmetrix Procedure Generator* is referenced for critical details on FLM operations for each supported host and multpath type.

IMPORTANT

For information on supported operating systems, file systems, and logical volume managers, refer to the *EMC Federated Live Migration Simple Support Matrix*. For any recent updates on FLM refer to *EMC Solutions Enabler Release Notes.*

All documents mentioned above are available at `http://support.EMC.com.`

A FLM configuration includes the network, storage arrays, application hosts, and the Solutions Enabler host, along with the required zoning and masking (Figure 27).

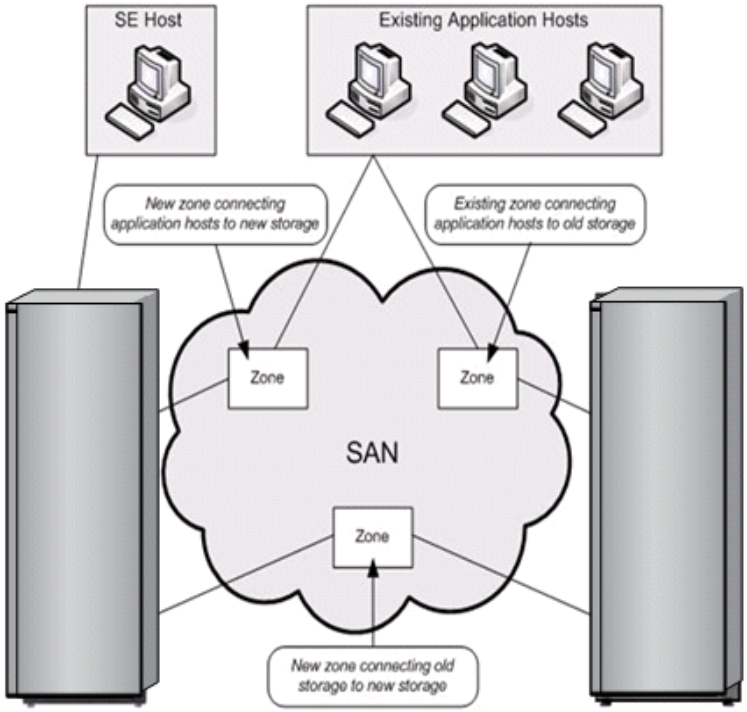


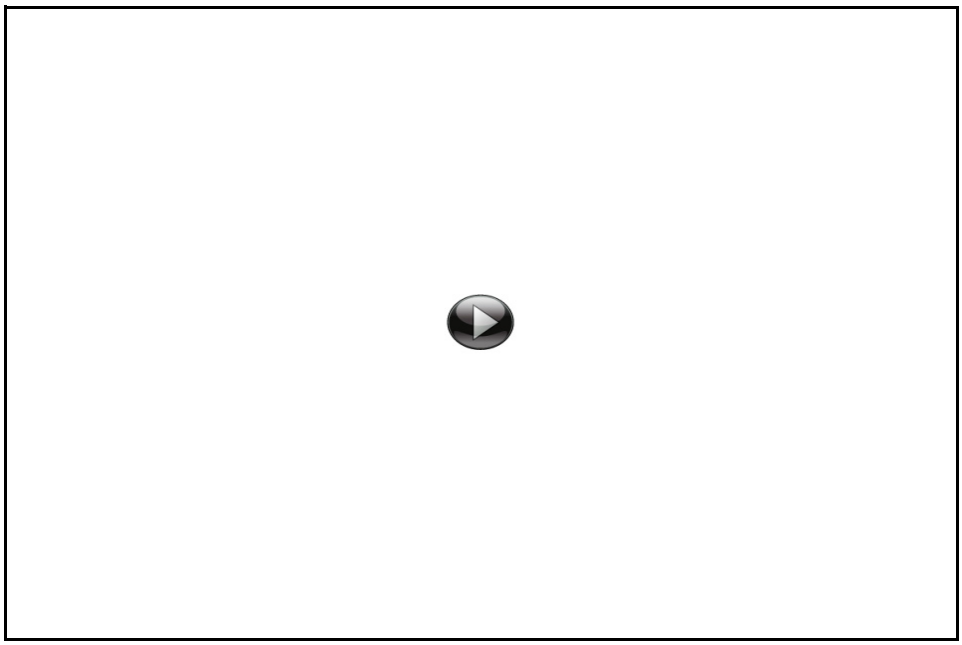
Figure 27 Example FLM configuration

The FLM process

The FLM process, as shown in the short animation in Figure 28, begins with verifying that the donor array is zoned to the application host. The donor array is then zoned to the new array, and the new array is zoned to the application host.

When the FLM session is created, a device file is used to pair devices from the donor array to the new array. During the create session, the new array devices assume the identity of donor array devices. The donor array remains active to the host and the new array is passive to the host.

When the FLM session is activated, the donor array devices are put into host_passive mode and the new array device into host_active mode. The host sees a new path for the donor devices. The data tracks are copied from the donor array devices to the new array devices using a hot, pull with donor update enabled. Donor update maintains a consistent image between devices in the event of a failed session.



To replay click the Rewind or Back

Figure 28 FLM process animation

Environmental and system requirements for FLM

FLM sessions require specific system prerequisites and preparation to the SAN environment, prior to creating and activating a session, and there are some array and device limitations as listed below.

Array/device requirements

FLM requires the following Enginuity versions and array/device configuration:

- Enginuity 5876 on the new arrays.
- Enginuity 5671 or 5773 on the donor DMX array and Enginuity 5875 or 5876 on the donor VMAX 20K array.
- The new array device must be the same size or larger than the donor array device.
- The donor array must be a DMX or VMAX 20K array and the devices are subject to the same restrictions as Open Replicator for a hot pull operation. Before and during each session, make sure the donor device is not the target of local or remote replication, as the ORS pull session is not able to copy consistent data if new data is written to the donor device.

Array/device limitations

FLM has the following array and device limitations:

- A maximum of 8 paths to the new array devices and donor array devices.

- A maximum of 32 device pairs at one time per host.
- No donor devices used for host-based clusters.
- No load balancing across multiple FAs.
- Boot devices cannot be migrated.

This is not a complete list of FLM requirements and restrictions. For further requirements/restrictions, refer to the *EMC Federated Live Migration Simple Support Matrix* and the *EMC Solutions Enabler Release Notes* available at `http://support.EMC.com.`

Federated Tiered Storage support

There is limited support for Federated Tiered Storage devices with FLM operations as described below.

With Federated Tiered Storage (FTS), an external LUN (eDisk) is attached through the SAN to a VMAX Family array, and is used as an external back-end disk for the VMAX array. FTS requires a new front end disk director (DX) and the Enginuity 5876 or higher.

Adding an eDisk through the SAN to the new array provides the ability to migrate user data from the external storage into the donor array. There are two modes for presenting the Disk to the new array, *external provisioning* and *encapsulation*.

For FLM, externally provisioned devices are fully supported and encapsulated devices are not supported.

For more information on FTS refer to "Federated Tiered Storage" on page 301.

Configuring the application hosts

Each operating system on the application host must be configured for FLM.

For information on supported operating systems, file systems, and logical volume managers, refer to the *EMC Federated Live Migration Simple Support Matrix* available at `https://support.EMC.com.`

Note: FLM now supports VMware hosts.

Zoning and masking prerequisites

FLM does not automatically setup any of the required zones for the migration. This includes zones from the application host to the new array, and from the donor array to the new array.

Prior to performing a FLM, zoning and masking must be configured as follows:

⚠CAUTION

Due to requirements of multipath environments, you must correctly configure paths to the new array devices to maintain data availability. Refer to "SAN setup requirements" on page 594 for more information on zoning requirements. For more information on zoning and masking refer to "Masking Devices with Auto-Provisioning Groups" on page 311. Verify the application donor array is visible on the application host and error-free.

- Verify the donor array is zoned to the application and SYMAPI hosts.
- Setup/modify zoning as follows:

- New array to application host — for application access to the new array.
- New array to SYMAPI host — for control of the new array.
- New array to donor array — for application data transfer.

◆ Mask the gatekeepers of the new array to the SYMAPI host, and mask the donor devices to the new array FA port(s).

Using FLM to perform a migration

FLM uses the SYMCLI `symrcopy` command to perform a migration session, The main control operations required to successfully complete a migration session are as follows:

◆ Create the session.

◆ Activate the session.

◆ Terminate the session.

Note: This section is an overview of the FLM process and the `symrcopy` operations, and is not a complete list of all steps required to successfully perform a FLM. For detailed steps for each supported operating system, refer to *EMC Symmetrix Procedure Generator* available at `https://support.EMC.com`.

Creating a device file

A device file is required to perform a FLM session. The device file specifies the source (donor) and target (new) devices using the format:

```
symdev=<symid>:<dev> symdev=<symid>:<dev>
```

The following is an example device file:

```
## FLM DEVICE PAIR FILE
## column1:target column2:source
symdev=000194900307:0171 symdev=000190100860:027A
symdev=000194900307:0172 symdev=000190100860:027B
symdev=000194900307:0173 symdev=000190100860:027C
symdev=000194900307:0174 symdev=000190100860:027D
```

Creating a session

To begin an FLM session, run the `symrcopy create` command and specify the `-migrate` and `-pull` options. The `-migrate` option indicates that this is a FLM operation, and this option must be used on all subsequent FLM operations.

The following is an example `create` command with the `-migrate` option:

```
symrcopy -f win_flm create -pull -migrate -host_type windows
-hba_type Qlogic -mp_type PPath_45 -v
```

Where:

◆ `win_flm` is the device file name.

◆ `Windows` is the application host platform type.

- `Qlogic` is the HBA type.
- `PPath_45` is the multipathing type.

Note: For the `symrcopy create` command, `-mp_type`, `-host_type`, or `-hba_type` are required for certain operating system types. For specific OS requirements, refer to *EMC Federated Live Migration Simple Support Matrix* available at `https://support.EMC.com`.

When a FLM session is created, the new array device assumes the WWN and geometry of the donor array device. The new array device is put in host_passive mode, so the donor array device is still accessed by the application hosts for reads and writes.

Prior to running the `symrcopy create` command, the geometry of the new array device can be changed using the `symconfigure` command. If this is not done, the geometry will be changed by the `create` command. While the migration session is active, the geometry cannot be changed using the `symconfigure` command.

IMPORTANT

The `-migrate` option must be used with the `activate, terminate, failback`, `set frontend_zero off,` and `set pace` commands when these commands are directed at the FLM session. If the `-migrate` option is used for any command for a session that was not created with this option, the operation is rejected. Refer to "Setting the session pace" on page 611 for more information on the `set pace` command.

Note: If the session creation fails at any point, the system is left in the same state prior to the `create` command.

Verifying the created session

Use the `symrcopy query` command to verify that the FLM pairs are listed as migration sessions in the Created state, as follows:

```
symrcopy -f win_flm query
```

```
Device File Name       : win_flm

       Control Device                   Remote Device      Flags      Status     Done
---------------------------- ---------------------------- ------- -------------- ----
                    Protected
SID:symdev          Tracks      Identification           RI CDSHUTZ  CTL <=> REM    (%)
------------------ --------- ------------------------ -- ------- -------------- ----
000194900307:00171      30000 000190100860:0027A        SD ...XXM.   Created       N/A
000194900307:00172      30000 000190100860:0027B        SD ...XXM.   Created       N/A
000194900307:00173      30000 000190100860:0027C        SD ...XXM.   Created       N/A
000194900307:00174      30000 000190100860:0027D        SD ...XXM.   Created       N/A

Legend:
R:  (Remote Device Vendor Identification)
  S = Symmetrix, . = Unknown.

I:  (Remote Device Specification Identifier)
  D = Device Name, W = LUN WWN, World Wide Name.

Flags:
(C): X = The background copy setting is active for this pair.
```

```
     . = The background copy setting is not active for this pair.
(D): X = The session is a differential copy session.
     . = The session is not a differential copy session.
(S): X = The session is pushing data to the remote device(s).
     . = The session is pulling data from the remote device(s).
(H): X = The session is a hot copy session.
     . = The session is a cold copy session.
(U): X = The session has donor update enabled.
     . = The session does not have donor update enabled.
(T): M = The session is a migration session.
     R = The session is a RecoverPoint session.
     S = The session is a standard ORS session.
(Z): X = The session has front-end zero detection enabled.
     . = The session does not have front-end zero detection enabled.
(*): The failed session can be reactivated.
```

Restoring new array device to host-active mode

If a created session is terminated before it is activated, the new array device can be restored to host-active mode. Use the following syntax to restore a new array device to host-active mode:

```
symdev host_active -sid SymmID SymDevName
```

Activating a FLM session

When a FLM session is activated, the donor array device is put into host_passive mode and the new array device into host_active mode. The host access is now directed to the new array device. Any device reservations are transferred from the donor array device to the new array device, and the tracks copy from the donor array device to the new array device.

IMPORTANT

Do not proceed with the `activate` command unless all the new array devices have been configured as new paths to each of the donor array devices involved in FLM. The next step in the process will set the donor array devices to host_passive. The new array devices must be masked, discovered, and configured into the multipath configuration on the application host, otherwise the activate command will be rejected. For details on performing these steps, refer to *EMC Symmetrix Procedure Generator* available at https://support.EMC.com.

The following is an example `activate` command with the `-migrate` option:

```
symrcopy activate -f win_flm -migrate
```

Note: The `-symforce` option can be included with a FLM `activate` command. This forces the `activate` operation to execute when it would otherwise be rejected. Use caution when using this option.

After the session has been successfully activated, tracks are copied from the donor array device to the new array device.

If the activate fails, the session will remain in the created state. If an error occurs after a session has been activated, the new array device is set to host_passive mode and the donor array device is set to host_active mode. All host access is directed back to the donor array.

Note: Any existing snap, clone, or RDF session on the donor array device will not be migrated.

Setting the ceiling value

The `symqos -rcopy set ceiling` command sets the maximum allowed bandwidth percentage for a given director, port, director/port pair, or all directors and ports. Valid values are 0 - 100 (%), or NONE (shuts off the ceiling function).

The recommended ceiling value for the new array is:

- 20% if the donor array is running Enginuity 5671
- 40% if the donor array is running Enginuity 5773

Note: Ceiling values may need to be adjusted to optimize the performance of the specific SAN environment.

The following example shows how to set a bandwidth ceiling of 40% for all directors on array 307:

```
symqos -rcopy set ceiling 40 -dir all -sid 307
```

The new ceiling setting is viewed by using the following `list ceiling` command:

```
symqos -rcopy list ceiling
```

Note: Setting the ceiling to a value (other than NONE) renders the session pace value ineffective to the copy. If the ceiling value is set to NONE, the session pace is in effect for the copy. For more information refer to "Setting the session pace" on page 611.

Monitoring FLM session status

A FLM session status is checked using the `symrcopy query`, `symrcopy verify`, or `symrcopy list` command.

Querying a session status

The following is an example session status query:

```
symrcopy query -f win_flm
```

```
Device File Name      : win_flm

       Control Device                  Remote Device            Flags     Status     Done
---------------------------- --------------------------------- ------ -------------- ----
                    Protected
SID:symdev          Tracks      Identification                 RI CDSHUTZ CTL <=> REM    (%)
------------------- --------- ------------------------------- -- ----- -------------- ----
000194900307:00171      30000 000190100860:0027A              SD ...XXM. CopyInProg     N/A
000194900307:00172      30000 000190100860:0027B              SD ...XXM. CopyInProg     N/A
000194900307:00173      30000 000190100860:0027C              SD ...XXM. CopyInProg     N/A
```

```
000194900307:00174      30000 000190100860:0027D              SD ...XXM. CopyInProg     N/A


Total                ---------
  Track(s)              120000
  MB(s)                  80062.5
```

For the output legend, refer to page 685.

The output from the `summary query` command is an abbreviated listing that shows all possible session states and the number of sessions that are in each state.

The following is an example session status query with the `-summary` option:

```
symrcopy query -f win_flm -summary
```

```
Device File Name : win_flm

  RCopy Session State           Count
  -----------------------      ------
  CreateInProg                      2
  Created                           0
  RecreateInProg                    0
  Recreated                         0
  CopyInProg                        3
  CopyOnAccess                      0
  CopyOnWrite                       0
  Copied                            1
  SyncInProg                        0
  Synchronized                      0
  Restored                          0
  RestoreInProg                     0
  Precopy                           0
  TerminateInProg                   0
  Failed                            0
  Stopped                           0
  FailedBack                        0
  VerifyInProg                      0
  Invalid                           0
  -----------------------      ------
  Total                             4

                      Track(s)    MB(s)
                     -----------  -------
                      120000      80062.5
```

Verifying a session status

The following is an example session verify status:

```
symrcopy -f win_flm verify
```

None of the device(s) in the list are in Copied state.

Note: If no verify option is provided, then Copied is the default state that is verified. Refer to "Verifying session state" on page 615 for more information on verify options.

The following example shows the output result when the `verify` command is used with the `-summary` option:

```
symrcopy -f win_flm verify -summary
```

Note: Note: 4t5The one-line `verify` command output comes after the `-summary` output.

```
Device File Name : win_flm

RCopy Session State           Count
  -----------------------      ------
  CreateInProg                      0
  Created                           0
  RecreateInProg                    0
  Recreated                         0
  CopyInProg                        4
  CopyOnAccess                      0
  CopyOnWrite                       0
  Copied                            0
  SyncInProg                        0
  Synchronized                      0
  Restored                          0
  RestoreInProg                     0
  Precopy                           0
  TerminateInProg                   0
  Failed                            0
  Stopped                           0
  FailedBack                        0
  VerifyInProg                      0
  Invalid                           0
  -----------------------      -------
  Total                             4

                                  Track(s)    MB(s)
                                -----------  -------
Total Protected                   120000      80062.5

None of the device(s) in the list are in 'Copied' state.
```

Note: If no verify option is provided, then Copied is the default state that is verified. Refer to "Verifying session state" on page 615 for more information on verify options.

Listing a FLM session

The `symrcopy list` command lists all migration sessions for a specific array. This command can be filtered to list only FLM sessions using the `-type <migrate|standard|RecoverPoint>` option.

The following is an example `list` command for FLM sessions only:

```
symrcopy list -sid 307 -type migrate
```

```
Symmetrix ID: 000194900307

Control Device            Remote Device             Flags      Status     Done
--------------------- ---------------------------------- ------ -------------- ----
        Protected
Sym     Tracks                Identification          RI CDSHUTZ    CTL <=> REM  (%)
-----   ------------ -------------------------------- -- ------ -------------- ----
00171        30000    60060480000000000619053594D314638 SW ...XXM.     Copied     N/A
00172        30000    60060480000000000619053594D314637 SW ...XXM.     Copied     N/A
00173        30000    60060480000000000619053594D314642 SW ...XXM.     Copied     N/A
00174        30000    60060480000000000619053594D314646 SW ...XXM.     Copied     N/A
```

```
Total ---------
  Track(s) 120000
  MB(s)     80062.5
```

For the output legend, refer to page 685.

Note: For listing all session types or only listing RecoverPoint sessions, refer to "Listing all sessions" on page 611 and "Filtering session types" on page 612.

Failing Back

Session failback can occur automatically or it can be performed manually. The following section describes automatic and manual failback of a FLM session.

Automatic failback

If an error occurs after activating a session, the session automatically fails back and the following actions occur:

- The new array device is set back to host_passive mode.
- The donor array device is set back to host_active mode.
- All host I/O is directed back to the donor array.

Manual failback

A migration session can be canceled anytime after the session is activated, which results in full data access to the donor array device.

Use the `failback` command with the `-migrate` option to cancel an activated session. The `failback` command can be run anytime following session activation, or when the copy is complete.

Running the `failback` command:

- Stops the FLM session
- Restores the donor array device to host_active mode
- Restores the new array device to host_passive mode.
- Does not restore the new array device identity. New array device still maintains the WWN and geometry of the donor array device.
- Restores any existing device reservations to the donor array device.

The following is an example `failback` command with the `-migrate` option:

```
symrcopy failback -session_name sym1_to_sym2 -migrate
```

Note: Automatic or manual failback will leave a session in the "FailedBack" state, and the only subsequent operation allowed is a `terminate` command.

Terminating a FLM session

After the copy session is complete or if a session has failed, use the `terminate` command with the `-migrate` option to end the FLM session.

Running the `terminate` command:

- Does not restore the new array device identity. New array device still maintains the WWN and geometry of the donor array device.
- Does not effect the host_active/passive mode setting of the donor array device. If the `terminate` command is run after a failback the donor array device is in host_active mode. If the `terminate` command is run after a session completes, the donor array device remains in host_passive mode.
- Keeps any device reservations with the new array device.

The following is an example `terminate` command with the `-migrate` option:

```
symrcopy terminate -session_name sym1_to_sym2 -migrate
```

Restoring new array device identity

The host may run indefinitely with federated identity on the new array devices; however, EMC recommends that the original array device identity be restored as soon as possible following a migration. Leaving the identity spoofing in place long-term has the potential to cause confusion for systems administrators or other users who may not be familiar with the details of FLM and how device identities are federated. This recommendation is provided only to raise awareness and is not intended as a mandate for unspoofing. New array devices may remain federated indefinitely and there is no requirement to unspoof at any time.

IMPORTANT

If the donor array device will be reused and is on the same SAN as the new array device, then the new array device must be restored to its original identity.

Refer to the *EMC Symmetrix Procedure Generator* available at `https://support.EMC.com` for details on restoring original device identity.

Restoring donor device to host-active mode

After the migration session is terminated and the new array device original identity is restored; or if the new array and donor array device are not on the same SAN, the donor array device can be restored to an active state using the `symdev host_active` command.

However, to avoid a situation where the donor array device and the new array device are both available to the host with the same identity, when `symdev host_active` command is issued the following conditions determine whether the operation is allowed to proceed or not:

- For any FLM supported Enginuity version, if either of the following conditions exist when the `symdev host_active` command is issued, the operation is blocked:
  - The device is mapped to an FA with VCM/ACLX enabled and masking records exist.

- The device is mapped to an FA with VCM/ACLX disabled.

◆ If either of the following conditions exist when the `symdev host_active` command is issued, the operation is allowed:

- The device is mapped to an FA with VCM/ACLX enabled, but no masking records exist.
- The device is not mapped to any FA.

Use the following syntax to restore a donor array device to host_active mode:

```
symdev host_active -sid SymmID SymDevName
```